

新手上路

——RoboCup3D 仿真环境建立和使用简介

(version 0.4)



东南大学 Robocup 仿真组
2006-11-08

目 录

第 一 节	RoboCup3D 简介	1
1.1 server		1
1.2 Monitor		8
第 二 节	如何建立 RoboCup3D 运行环境.....	8
第 三 节	开始一场比赛.....	10
第 四 节	SEU-3D 程序简介	11
附录		12
附录 A rcssserver3D. 0.3 参数		12
附录 B rcssmonitr3D-lite 使用说明		14
附录 C RoboCup 3D 仿真比赛成绩排名		15
说明		17

第一节 RoboCup3D 简介

RoboCup (Robot World Cup)，即机器人世界杯足球锦标赛。它是国际上的一项为提高相关领域的教育和研究水平而举行的大型比赛和学术活动，通过提供一个标准任务来促进分布式人工智能、智能机器人技术、及其相关领域的研究与发展。

RoboCup 分为仿真组，中型组，小型组，四腿组和类人组。分别进行不同技术体系的研究，以加快人工智能的发展。

RoboCup的目标是在2050年之前，建立一只能够打败人类足球世界杯冠军队的机器人球队。

2004年3D足球仿真比赛诞生，新的比赛加入了第三维使得比赛更加真实。3D足球仿真器只是0.5.2版，还远远没有完善，而相关文档也很少。

3D仿真更加接近人类比赛环境，研究重点是球员的感知、基本动作及其决策、与服务器的通信等基本功能模块，以实现队员在3D环境中的实时响应。3D的与2D的有类似的地方，但比赛环境和物理模型有很大的差别，目前各个高校都处在对比赛环境和基本动作的研究中，一旦比赛环境完善、底层成熟之后，中上层的策略将可以移植或借鉴2D仿真研究的许多成果。

为了开展研究和设计仿真队伍，首先必须先熟悉仿真系统的组件。3D 足球仿真系统包含 3 个重要的部分：server、monitor 和 agent。除了要设计的 agent，对 server 和 monitor 也要熟悉。

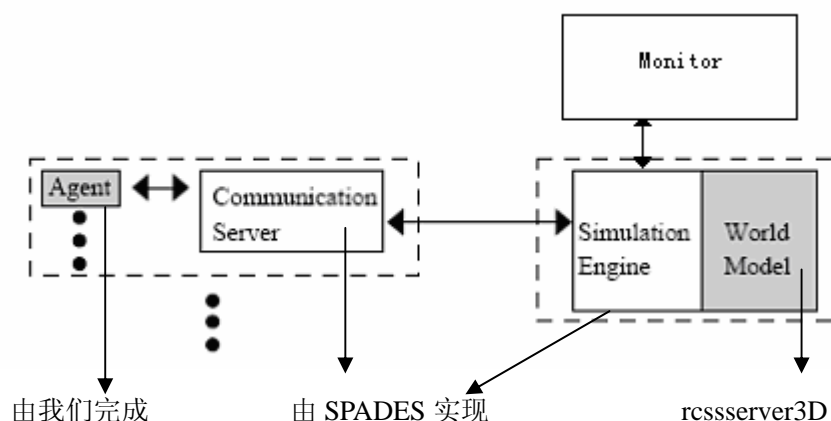


图 1-1RoboCup3D 仿真系统组件

1.1 server

为了能够在 server 上工作，首先必须了解仿真中间系统 SPADES (System for Parallel Agent Discrete Agent Simulation)。

1.1.1 SPADES

仿真 server 是在 SPADES(System for Parallel Agent Discrete Agent Simulation)

平台上实现的。SPADES 是分布式仿真的中间系统(Riley 2003)。其主要目的是为运行多计算机仿真系统提供一个一般性的平台。它实现了 agent 和仿真世界之间的交互，而用户不需要担心接口、地址等等。

SPADES 主要特点是：

- agent 以执行为基础——支持实现感觉，思考和动作。
- 分布式计算——支持在多台计算机上运行 agent 程序。
- 结果不受网络延时或者机器间负载不同的影响——SPADES 确保时间按照恰当的顺序进行。
- agent 可独立地从编程语言编程——agent 可以用任何语言编写，只要支持读/写 PIPEs。
- 动作不需要同步——agent 的动作可以在仿真的不同时间执行。

(一) 组织结构：

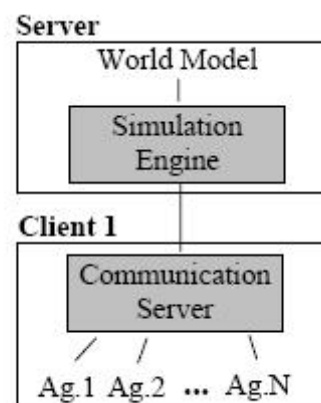


图 1-2 SPADES 结构图

SPADES 结构是 c/s 结构(图 1-2)。Simulation Engine 和 Communication Server 作为 SPADES 的一部分提供；当使用者建立了 agent 和 World Model 就可以运行在前者上了。Simulation Engine 是允许在其上建立特殊 World Model 的一般软件。它运行在 server 端，并通过 Communication Server 为 agent 和 server 提供交互和通讯。Server 的 World Model 必须运行在同一台机器上。作为 client 分布的是 agent 和 Communication Server。Communication Server 必须出现在所有 client 的机器上来提供 agent 和 Simulation Engine 的通讯。它从 agent 接收信息并发送到 server。agent 也运行在 client 端。

(二) 感觉—思考—动作 周期：

SPADES 实现感觉—思考—动作 周期，每个 agent 接受感觉信息并返回动作信息。这就意味着 agent 只能在接收到感觉信息之后有动作。当然自身的感觉也

可以要求一个动作，但原理依旧——一个感觉产生一个动作。这样，SPADES 提供了一个动作叫做请求时间通报，返回一个空的感觉(时间通报)，agent 在接收到后可以反应一个动作。例如，如果 agent 接收到一个感觉在 100 周期并希望在 110 周期产生一个动作，但是下一个感觉只有在 120 周期才会到达，agent 可以要求在 110 周期接收到一个时间通报并在接收到之后发出命令。

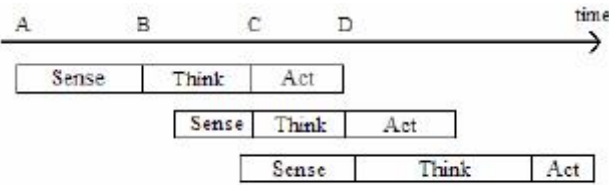


图 1-3 SPADES 感觉—思考—动作 周期

图 2-3 描述了感觉—思考—动作 周期和各个部分运行的时间。A到B是发送一个感觉到agent。接收到感觉后(B到C)agent决定执行哪些动作；然后(C到D)动作发给server。

在agent中，感觉—思考—动作部分是时间上重叠的（如在图 1-3 中）；但有这样一个限制——agent的思考周期是不重叠的。这个约束是有意义的，因为每个agent使用一个进程单位，因此，同时只有一个感觉被计算。

（三）使用 SPADES

与 2D 仿真不同的是：3D 比赛中由 server 负责启动 agent。SPADES 库调用包含有要启动的 agent 的数据库文件，该文件是./app/simulator/agentdb.xml 文件。Agent 通过 UNIX 管道连接到 SPADES 的 Commserver 端。Commserver 轮流与 server 通讯。默认设置是 server 和整合的 Commserver 一起启动。可以通过 Commserver 在不同的系统上分布运行 agent 进程。先启动 server，然后启动相应数目的 Commserver。相关的参数设置在 server 的设置脚本 rcssserver3D.rb 中。'Spades.RunIntegratedCommserver' 决定是否运行整合的 Commserver。'Spades.CommServersWanted'指 server 等待的 Commserver 数目。

1.1.2 ODE

见ODE用户指南。

1.1.3 server 的虚拟比赛环境

Agent 的活动空间是由 server 的虚拟环境决定的。

比赛场地是标准的 FIFA 足球场（长 100m~110m，宽 64m~75m）。球门和球也是 FIFA 标准：球门宽 7.32m，足球直径 0.222m，重 0.41kg~0.45kg。因为 agent 太小又不能跳，所以球门只有 0.5m 高，FIFA 标准是 2.44m。

FIFA 没有指定重力加速度的大小，但在仿真比赛中重力加速度定为 $9.81 \frac{m}{s^2}$ 。

仿真步为 0.01s。默认情况下 monitor 每 15 仿真步更新一次。这些值都是可以在server的配置文件./app/simulator/rcssserver3D.rb中可以设置的。

机器人直径是0.44m，重75kg。机器人是全向驱动的。当然，全向驱动只能在与地面接触的时候。最大速度目前限制为10 m/s。通过测试目前版本（rcssserver3D-0.52）中agent最大速度为1.6m/s。

1.1.4 server 与 agent 交互的信息格式

3D仿真server允许22个agent(每个队11个)与server交互来进行一场仿真机器人足球比赛。每个agent接收到关于和其他队员、球门等相对位置和其他与比赛有关信息组成的感觉。这时世界中各个物体的位置信息由360度视觉提供的是难于使用的信息。作为反应，agent发送一个动作，像drive或者kick。drive是要求一个给定方向的力作用在身体上，而kick意味着一个从agent出发的放射状的力作用在球上。server每周期发送一个感觉，每个周期200毫秒，10毫秒一个仿真步。

主要信息分为感觉和动作两大类。感觉就是server发送给agent的信息；而动作就是agent发送给server的信息。另外还有一些在通讯中很重要的格式。

（一）感觉

Agent现在能接收4种感觉：Vision, GameState, AgentState, hear。每种感觉信息以字母"S"开头，后面跟着两个整数(time)。第一个周期是感觉发出的周期；第二个周期是信息到达agent的周期。

感觉信息的格式是：

S*time time data*

data是感觉信息的字符串（注意：一条感觉信息可以包含多种感觉）。

（1）视觉

视觉提供场地中物体的空间排列。物体包含球员、足球、球门和角旗。在0.3版本中，视觉是全方向视觉并且物体是透明的(至少对视觉)。物体的位置以相对于agent的极坐标形式给出。坐标参数包含距离、水平方向的角度—theta 和垂直方向的角度—phi。

（Vison

(Flag (id id)(pol d theta phi))...

(Goal (id id)(pol d theta phi))...

(Ball (pol d theta phi)) ...

```
(teamname (id id)(pol d theta phi))...  
)
```

(2) 比赛状态

比赛状态提供与比赛有关的信息。它提供场地、球门、球和球员的尺寸。还提供物体的质量和比赛的其他信息：时间、比赛模式、球员号码、球员是左边还是右边的。格式是：

```
(GameState  
  (team side)  
  (unum number)  
  (FieldLength length)  
  (BallMass mass)  
  (playmode playmode)  
  ...)
```

(3) Agent状态

agent状态提供agent自身内部的信息。现在只提供电池状态和温度，而这两项参数目前对比赛还没有任何影响。

```
(AgentState  
  (battery battery)  
  (temp temp)  
)
```

(4) 听觉

从0.4版本开始，agent可以使用听和说来互相通讯。

```
(hear dir msg)
```

这里 *dir* 表示听觉信息传来的方向角度，如果是self表示该信息是自己say的。*msg*为say的内容。

听觉的限制范围为50m，在当前配置中，每个周期至多只能听到每个队伍的一条信息，如有多人说话，听到谁的信息由动作执行的先后顺序决定。

(二) 动作

与感觉一样，通过一些动作来实现agent对环境的作用。每种动作信息以字母"A"开头，后面是字符串。

```
Adata
```

data 包含动作的信息。

Create

这是agent必须发送的第一个动作。这个动作使得server登录agent并创建与agent的通讯。格式如下：

A(create)

Init

这个动作使得server接收agent的详细信息：号码和队名。如果号码是0，server会自动地给agent一个号码。格式如下：

A(init (unum *number*)(teamname *name*))

Beam

这个动作使得在比赛之前agent传送到一个给定位置，而不要遵守物理定律。格式如下：

A(beam *x y z*)

x y z 为球场的全局坐标的位置值

Drive

这个动作使得agent移动。这个动作在agent身体中心产生一个力(*x,y,z*)，最大力量为100.0。格式如下：

A(drive *x y z*)

x y z 为以agent中心为起点的力量（矢量）

Kick

在有重力并遵守物理运动定律的情况下，agent可以用kick动作以给定的力量和方向踢球。垂直方向作为一个参数—垂直方向角。水平方向是以agent身体呈放射状的。这个动作必须这样发送：

A(kick *angle force*)

angle 为踢球力量的竖直方向的角度

force 为踢球力量的大小，最大为100

Catch

只有守门员（必须为1号）才能执行该动作。这个动作使得在一定距离（0.8m）守门员身体的球被守门员抓住，同时将守门员周围（距离小于2m）的其他队员弹开；如果球不在可catch范围内，则该动作被忽略。格式如下：

A(Catch)

Say

Agent执行说话动作。*msg*为说话的内容，必须为不包含空格、圆括号、反斜杠的小于512字节的字符串。格式如下：

A (Say *msg*)

(三) 其他重要的通讯格式

server发送 **done("D")**信息表示创建通讯。agent接收到这条信息后必须执行初始化程序，完成初始化后必须向server发送 **initdone("I")** 信息。然后，server开始发送感觉信息，agent返回动作。每个动作必须以 **done('D')**结尾(图1-4)。

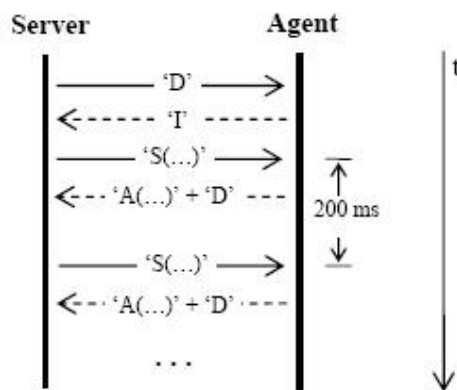


图1-4 agent和server通讯时间表

如前所述，每20个server仿真步接收到一个感觉信息，所以agent只能在20个仿真步内执行动作。然而，不仅仅如此，因为agent可以要求server在指定的周期发送一个 **request time notify(R)** 感觉信息。信息格式如下：

Rtime

time是server必须回复的周期。这个过程使得server在指定周期返回一个空的感觉信息('T')。如前所述，这样的目的是在两个感觉信息之间的指定时间接收空的感觉信息来执行动作。实验证明如果早于当前周期发送request time notify，server允许任意执行动作。信息格式如下：

Ttime

time仍是server回复的周期。

使request time notify有意义的例子是：当球员想跳起来碰空中的球的时候。因为两条感觉信息之间有200毫秒，有可能周期t时球离agent太远，而周期t+1时球已经传过了。为了越过这个障碍，agent可以预计一个起跳时间，并要求接收一个time notify 信息，实现在恰当的时间实现正确的动作。

C

agent向commserver发送C信息,要求commserver恢复agent的当前思考时间,commserver在收到C信息之后,回复Ctime, time就是agent的思考时间。

1.2 Monitor

Monitor是用来观看比赛和播放log的,通过monitor你可以看到agent的表现。默认的monitor是rcssmonitor3D-lite (操作说明见附录),目录为./app/rcssmonitor3d/lite。连接server使用参数为: --server <serverIP>;播放log使用参数: --logfile <filename>。server记录的日志文件夹Logfile/将在你启动server的目录下产生。

另外, monitor协议支持训练功能,可以重现一个特定的测试场景。这个功能在后面的trainer中介绍。

第 二 节 如何建立 RoboCup3D 运行环境

2.1 安装Linux系统

由于比赛完全是在Linux平台上运行的,所以第一步是安装Linux系统。由于Linux的发行版本很多,各个发行版本的安装不完全相同,但是安装都有图形界面,比较容易。这里说明一些重要步骤:

分区:

分区	/	swap	boot	/home
中文名	根分区	交换分区	启动分区	用户目录
分区格式	ext2等	swap	ext2等	ext2等
分区大小	>2G	=内存大小	10M左右	
功能	储存文件	相当于虚拟内存	储存启动文件	用户文件目录
备注	boot分区不是必需的,启动文件可以储存在根分区/boot文件夹下。 /home分区也不是必需的,如无该分区,则文件存储在根分区/home文件夹下			

选择安装软件包:

为了防止缺少运行时需要的软件包,在硬盘空间允许的情况下,这里我们推荐**完全安装**!如硬盘空间紧张也要将**开发相关的软件全部选上**。

防火墙设置: **关闭防火墙**,避免网络运行的问题。

2.2 安装rcssserver3d

下面以在Susu10.1上安装rcssserver3d为例,其他Linux发行版操作也类似。

(i) 使用自动安装包 rcss-suse-install 安装

将rcss-suse-install安装包解压缩，直接运行安装包中的setup.sh（如不能直接运行，请在终端中运行），按照提示输入密码即可。

(ii) 手动安装

下面是从源码安装3D仿真平台的方法。在安装rcssserver3d之前，我们还需要安装一些必需的软件包。下面的安装说明中蓝色字在终端中运行的命令。

1. 安装boost-1.33.1

```
# cd tools/build/jam_src
# ./build.sh
# cp bin.linuxx86/* ../.././
# cd ../.././
# ./bjam "-sTOOLS=gcc" install
boost的编译安装时间也较长
```

2. 安装ode-0.5

```
# make configure
# make ode-lib
# make drawstuff-lib
# make ode-test
# make drawstuff-test
# cp -a include/ode /usr/local/include
# cp lib/libode.a /usr/local/lib
```

3. 设置Java环境变量

```
# export JAVA_HOME=/usr/share/java
# export CLASSPATH=.:$JAVA_HOME/lib/dt.jar:$JAVA_HOME/lib/tools.jar
# export PATH=$PATH:$JAVA_HOME/bin
```

4. 安装spades-1.10

```
# ./configure
# make
# make install
```

5. 安装ruby-1.8.3

```
# ./configure --enable-shared
# make
# make install
```

6. 安装 rcssserver3d-0.5.2

```
# export CPPFLAGS=-I{$boostdir}\-L/usr/local/lib
# ./configure
如果希望安装bipped(双足仿真机器人)需要加上参数—enable-kerosin
# make
# make install
配置环境变量
在~/.profile中添加一行:
LD_LIBRARY_PATH=/usr/local/lib/rcssserver3d:$LD_LIBRARY_PATH
```

至此rcssserver3d安装完成，可以运行rcsoccersim3D，启动默认的agenttest来检查安装成果。如需要运行bipped仿真，需要分别进入bipped相应的server,agent和monitor目录分别运行。

第 三 节 开始一场比赛

如上一节所介绍的，你可以通过 rcsoccersim3D 启动 rcssserver3d 和 rcssmonitor3D-lite。也可以单独运行 rcssserver3d 命令启动 server。Server 的配置文件在\$HOME/.rcssserver3d 目录下。部分配置参数在第一节中已经介绍。这里要介绍的是 agentdb.xml 文件。

agentdb.xml 指定了 agent 的目录和运行参数。SPADES 通过 agentdb.xml 来查找和启动 agent 队伍。默认的情况下 SPADES 会调用当前目录下的 agentdb.list 来作为 agentdb.xml。当然你也可以通过参数--agentdb_fn <agentdb.xml>指定 agentdb.xml 文件。下面我们来看一下 agentdb.xml 文件。

```
<agent_type_external name="SEU-3D">
  <inputfd>3</inputfd>
  <outputfd>4</outputfd>
  <timer>jiffies 10000</timer>
  <working_dir>/home/3d/SEU-3D</working_dir>
  <exec_line>seu-3d-player</exec_line>
</agent_type_external>
```

第一行中的 name 与 rcssserver3D.rb 中的名字对应:

```
# queue agents for startup
spadesServer.queueAgents('SEU-3D',11)
```

即 SPADES 按照 rcssserver3D.rb 中要启动的 name 和启动 agent 的数目启动,而 name 所对应的可执行码在 agentdb.xml 中指定.

第二行和第三行分别指 agent 的输入和输出管道的号码.

第四行是指定计时器类型和每个周期允许运行的指令条数.

第五行是 agent 的运行目录;

第六行是启动 agent 所需要执行的命令

第 四 节 SEU-3D 程序简介

SEU-3D 分为 3 个层次：控制层、技术层和接口层，另外还有其它辅助模块。每个层次中又按功能分为不同的模块。

控制层：

Player 模块包含 agent 的中上层动作和球队策略。含有在某个特定情况下用来推理 Agent 应该执行的最佳动作的函数。程序中是根据 World Model 中最新的信息和 agent 在当前阵型中所起的作用来选择要执行的动作的。BasicPlayer 是 Player 的基类,而 BasicPlayer 又是从 Agent 类继承而来的。

Formations 中包含了不同的球队阵型信息，存储当前阵型信息，和 agent 在阵型中的号码用来决定 agent 的角色。更进一步，它包含了一个根据当前阵型和球的位置决定 agent 在场上策略位置的函数。

Settings：这个组件中包含了 player 参数的信息（这些信息会影响 agent 的决策过程）和 server 参数的信息。Settings 中的大部分参数是阈值参数，用来决定 agent 是否应该执行某个动作。

技术层：

WorldModel 模块是维持 World Model 的更新。这些信息包含：球的位置，其他人(和自己)的位置，球门的位置，甚至是比分、时间信息也存储在这个模块。其中 BasicWorldModel 是 WorldModel 的基类。

Skills 模块包含了 Agent 执行个人动作的必要信息（例如，截球，或将球踢到某个特定位置）。

接口层：

commserver 模块完成简单地从 server 接收/发送信息的任务，Agent 模块构成发送的信息。

其他辅助模块：

Object：这个组件描述了所有对象的信息：速度、位置、质量等。分为 FixedObject(固定物体), DynamicObject(运动物体),PlayerObject(球员),AgentObject(Agent 自己),BallObject(球).它们之间也是继承的关系。

Geometry 模块使得几何计算更加简单。用来计算距离、矢量等数据。

Log 模块作用是写 log 文件，用来调试程序。

编译说明：

首先将压缩包解压，然后执行

`./build`

即可。

修改代码之后只执行 `make` 即可

附录

附录 A rcssserver3D. 0. 3 参数

参数		数值
Agent 质量		75kg
Agent 半径		0.22m
Agent 最大速度		1.6m/s
Drive 最大力量		100
Drive 作用时间		直到改变
Drive 误差		每个坐标 2%
Kick 最大力量		100
Kick 竖直角度		0° ~50°
Kick 作用时间		100ms
Kick 半径		0.04m
Kick 误差 (正态分布)	水平角度	$\sigma = 0.02$
	竖直角度	$\sigma = 4.5$ (角度为 0° 或 50° 时 $\sigma = 0.9$)
	力量	$\sigma = 0.4$
Agent 运动方程		$14.43 \cdot a = 0.4848 \cdot F - 30 \cdot V$
球的质量		0.41kg~0.45kg
球的半径		0.111m
球与地面的摩擦系数		0.2148174
球与空气的摩擦 系数	水平方向	0.3
	竖直方向	0.3
球场长度		100m~110m
球场宽度		64m~75m
球场高度		40m
球门宽度		7.32m
球门高度		0.5m
球门深度		2m
Flag1_1 位置		(-FL/2,-FW/2)
Flag1_r 位置		(FL/2,-FW/2)
Flag2_1 位置		(-FL/2,FW/2)
Flag2_r 位置		(FL/2,FW/2)

Goal1_l 位置		(-FL/2,-GW/2)
Goal1_r 位置		(FL/2,-GW/2)
Goal2_l 位置		(-FL/2,GW/2)
Goal2_r 位置		(FL/2,GW/2)
重力加速度		$9.81 \frac{m}{s^2}$
球与地面碰撞后速度衰减		0.6
仿真周期		200ms
仿真步		10ms
进球后暂停时间		3s
Kick In 允许时间		1s
半场时间		300s
任意球隔离距离		9.15m
开球之前等待时间		2s
视觉误差	校准误差	-0.005m~0.005m
	距离误差	$\sigma = 0.0965$
	水平角度误差	$\sigma = 0.1225$
	竖直角度误差	$\sigma = 0.1480$

附录 B rcsssmoniotr3D—lite 使用说明

参数/命令	功能
rcsssmoniotr3D—lite	启动
--server <serverIP>	连接到 server
--logfile <filename>	播放 log 文件
--help	显示帮助信息
w 或 ↑	镜头向前
a 或 ←	镜头往左
s 或 ↓	镜头往下
d 或 →	镜头往右
c	镜头跟踪球或取消跟踪
k	开球
b	坠球
n	显示号码或不显示号码
p	暂停
r	运行仿真
q	Monitor 退出
2	显示小地图
以下命令为播放 log 时使用	
Shift+m	正常播放
>	单步播放

附录 C RoboCup 3D 仿真比赛成绩排名

2004 在葡萄牙里斯本 (Portugal Lisbon) 举行的国际比赛:		
1	Aria	Amir Kabir University of Technology, Iran
2	AT-Humboldt	Humboldt University Berlin, Germany
3	UTUtd2004	University of Teheran, Iran
德国公开赛2005:		
1	Brainstormers 3D	Germany
2	Caspian	Iran
3	RoboLog	Germany
美国公开赛 2005:		
1	UTUtd	University of Teheran, Iran
2	Caspian	Iran
3	MRL	Germany
4	Espadana	
日本大阪 RoboCup2005		
1	Aria	Amir Kabir University of Technology, Iran
2	Brainstormers 3D	Germany
3	ZJUBase	浙江大学, China
3	Caspian	Iran
7	WrightEagle2005	中国科技大学
2005 中国机器人大赛(常州)		
1	ZJUBase	浙江大学
2	WrightEagle2005	中国科技大学
3	Fantasia	大连理工大学
4	TsinghuAeolus	清华大学
7	SEU-legend	东南大学
2005 AI Games (Iran)		
1	Aria	Amir Kabir University of Technology, Iran
2	ZJUBase	浙江大学
3		
4	SEU-3D	东南大学
2006 Iran Open		
1	Aria	Amir Kabir University of Technology, Iran
2	ZJUBase	浙江大学

3	Caspian	Iran
3	Arman	
Top8	SEU-3D	
2006 Dutch Open		
1	FC Portugal	Portugal
2	SEU-3D	东南大学
德国不莱梅 RoboCup2006		
1	FC Portugal	Portugal
2	WrightEagle2005	中国科技大学
3	ZJUBase	浙江大学
4	Aria	Amir Kabir University of Technology, Iran
5	Rezvan	Iran
6	SEU-3D	东南大学
7	MRL	Iran
8	Virtual Wender	Germany Bremen University
苏州 China Open 2006		
1	Fantasia	大连理工大学
2	SEU-3D	东南大学
3	Aeolus	清华大学
4	Everst	北京理工大学

说明

- 0.4 版 [许元xuyuan.cn@gmail.com](mailto:xuyuan.cn@gmail.com) 更新rcssserver3d的安装，调整排版
- 0.3 版 [许元xuyuan.cn@gmail.com](mailto:xuyuan.cn@gmail.com) 更新rcssserver3d的安装、附录
- 0.2 版 [许元xuyuan.cn@gmail.com](mailto:xuyuan.cn@gmail.com) 根据rcssserver3d-0.5.2 更新server的通讯格式，更新附录
- 0.1 版 [许元xuyuan.cn@gmail.com](mailto:xuyuan.cn@gmail.com) 第一版，主要内容有 3Dserver简介，安装，运行以及SEU-3D简介

如果您对本文档有任何建议，请与我们联系。