

一种高级机器人语言中的视觉功能 及其引入方法

尹宝林

(北京航空学院)

摘 要

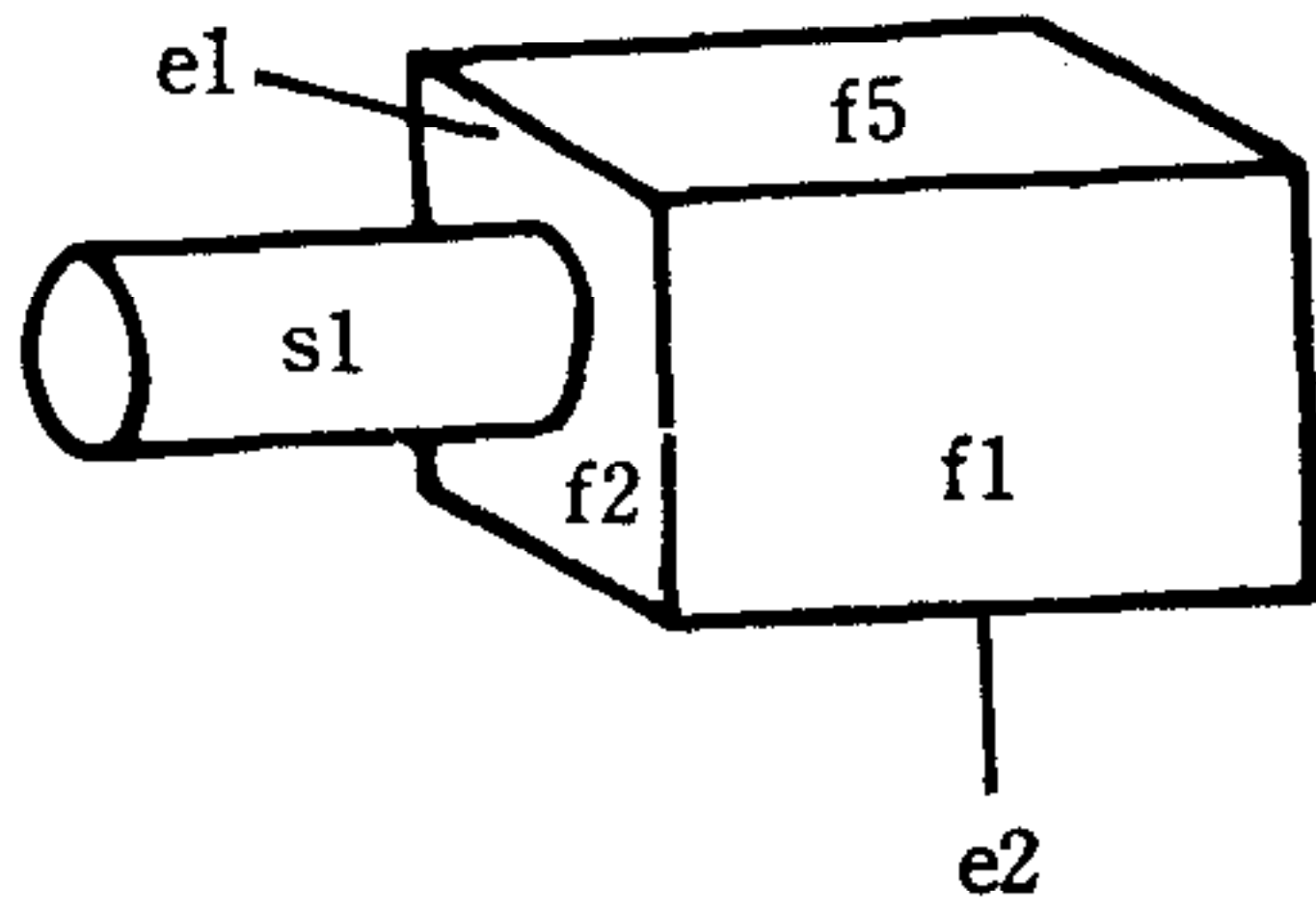
本文讨论了一种在高级机器人语言 RAPT 中使用视觉信息的方法。在这个方法中，视觉信息主要用来发现机器人工作环境中物体的预期位置与实际位置间的差异，并对机器人的动作进行相应的调整。符号推理系统使得大量的视觉信息处理工作得以在编译阶段进行，以保证在程序运行阶段视觉系统工作的实时性。视觉信息框架结构使得机器人系统可以充分地利用视觉信息，以更新其有关工作环境状态的信息。

在机器人的程序设计，即任务控制中使用传感信息来检测周围工作环境的状态，并据此调整机器人的动作，是使机器人克服工作环境中不确定性的影响，顺利完成工作任务的一个重要方法。要想成功地使用传感信息，必须具备两个条件：一是要有良好的传感器系统以提供准确的信息；二是要有合适的方法来处理和使用传感信息。本文着重讨论作者在高级机器人语言 RAPT 中使用视觉信息的方法。

一、RAPT 语言简介

RAPT 是英国爱丁堡大学从七十年代中期开始发展的一种用于机械装配的高级机器人语言。在 RAPT 中，机器人任务是以被机器人操作的物体的运动及相互的空间关系的形式来描述的。用户在程序中，不必考虑机器人手部装置在操作过程中每一步的具体位置和运动方向。这些信息以及其它必要的操作细节将由语言的编译系统根据用户程序和工作环境自动地推算出来，并进一步转换成机器人可以执行的命令。因此，RAPT 是一种物体级机器人程序语言。

RAPT 语言的描述功能可以分为两部分。第一部分是用来建立被操作物体模型的模型构造系统。在 RAPT 中，每一个物体都可以用其表面的几何特征来描述。这些几何特征包括顶点、边缘、平面、球面、柱面等。每一个物体的模型都指明该物体有哪些几何特征及其在该物体局部坐标系中的位置。用户不需要完整地描述一个物体，而只需要描述在有关程序中所要引用的特征即可。图 1 是一个物体及其 RAPT 语言的描述。



```

body/body1;
  p1 = point/0, 0, 0;
  p2 = point/20, 0, 0;
  p3 = point/20, -20, 0;
  p4 = point/20, -20, 30;
  p5 = point/20, 0, 30;
  p6 = point/0, 0, 30;
  p7 = point/0, -20, 0;
  p8 = point/10, -10, 0;
  p9 = point/10, -10, -20;
  f1 = face/p1, p2, p5, ylarge;
  f2 = face/p1, p2, p3, zsmall;
  f3 = face/p1, p6, p7, xsmall;
  f4 = face/p4, p5, p6, zlarge;
  f5 = face/p3, p2, p4, xlarge;
  l1 = line/p2, p3;
  l2 = line/p1, p6;
  l3 = line/p8, p9;
  e1 = edge/l1, ysmall;
  e2 = edge/l2, zlarge;
  s1 = shaft/axis, l3, radius, 5, zsmall;
  terbod;

```

图 1 一个物体的 RAPT 描述

RAPT 的第二部分功能是对装配任务中工作环境的起始状态、各个中间状态和最终状态以及动作的描述。在 RAPT 中，一个任务可以表示成工作环境状态的序列 $\{S_i | i = 0, 1, 2, \dots, N\}$ 。两个相邻的状态 S_i 和 S_{i+1} 是由动作来转换的。每一个状态可以由环境中各个物体的几何特征之间的关系来描述。对动作的描述可以明确地规定动作的类型、方向和幅度，也可以由动作前后的环境状态隐涵地规定动作的细节。RAPT 中不提供控制流语句，因此所有的动作语句按顺序执行。图 2 是一个简单的 RAPT 程序段的例子。为了全面描述装配任务，RAPT 中还提供了联结、部件、宏定义和宏调用等命令和功能。详细内容可见文献[1]。

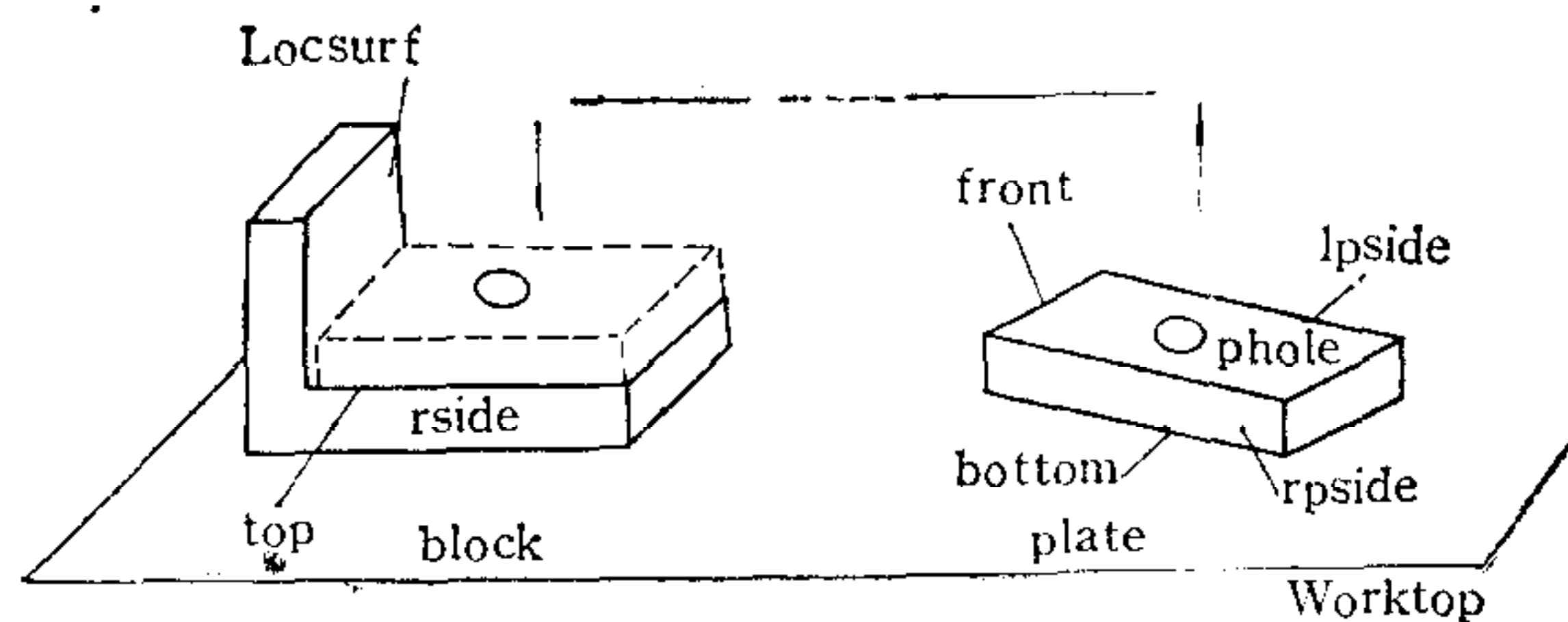


图 2 一个机器人任务的 RAPT 描述

```

call/pickup, plate,rpside of plate, lpside of plate, phole of plate, worktop;
  调用宏指令使机器人抓住 plate.
move/plate, perto, top of plate, by 10;
move/plate, parrel, top of plate; 将 Plate 放到 block 上.
move/plate, perpto, top of plate;
  against/rpside of plate, rside of block;
  against/bottom of plate, top of block;
  against/front of plate, locusurf of block;
Call/letgo; 调用宏指令使机器人离开.

```

二、RAPT 语言的编译和几何推理

RAPT 系统的工作分为两个不同的阶段：脱机编译和联机运行。在编译阶段，编译系统根据用户程序产生机器人控制命令序列。这一命令序列在运行阶段控制机器人的动作。

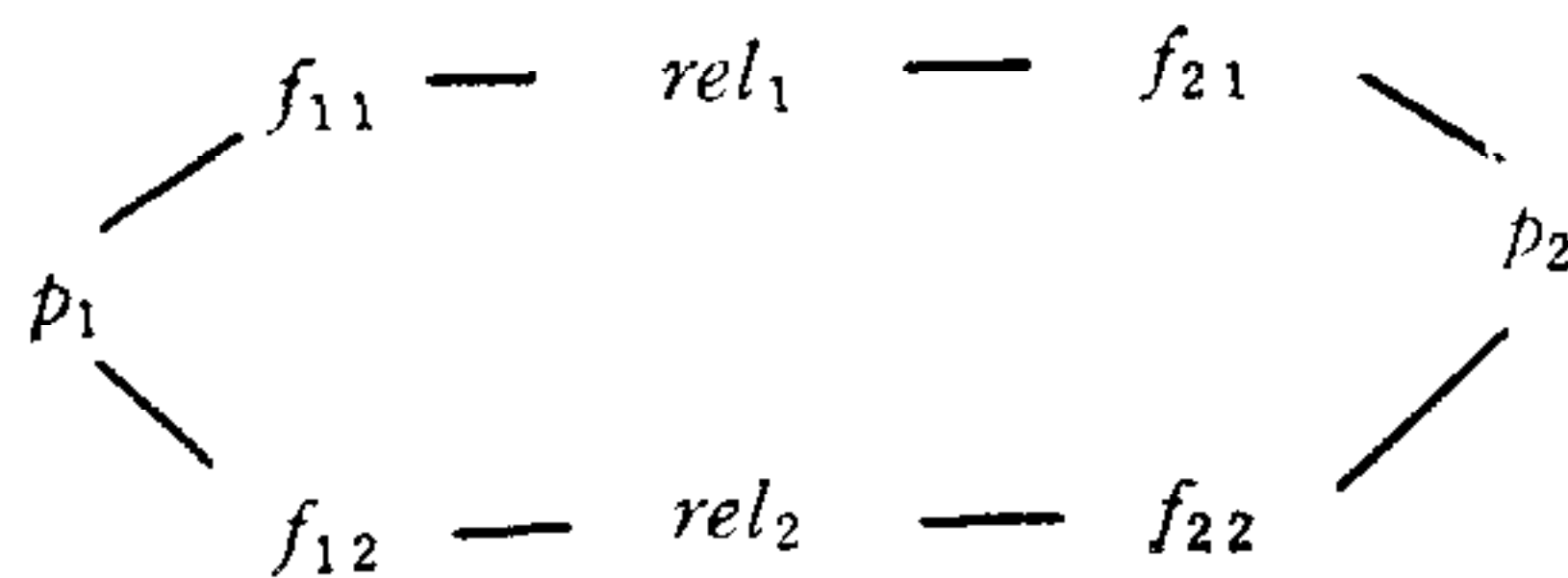
RAPT 的编译系统分为两个模块，即几何推理模块和后置处理模块。几何推理模块是 RAPT 编译系统的核心。在几何推理过程中，编译系统对用户程序进行分析和推理，以便确定在每一工作环境状态下各个物体在参考坐标系中的位置。这一工作过程是与目标机器人无关的。这些信息再由后置处理模块转换成能由目标机器人执行的命令。编译系统的这种组织方式可以灵活地适应于不同类型的目标机器人。下面简单介绍一下几何推理模块的工作原理^[2]。

RAPT 的编译系统首先将用户程序中的信息组织在一个关系网络中。在这个关系网络中，每一个关系都由一个四元组 (r, F_1, F_2, S) 表示。其中 r 表示关系的类型； F_1 和 F_2 分别是由 r 所联系的两个物体中的两个几何特征； S 是这一关系成立的环境状态。这一四元组描述了两个物体通过各自相应的几何特征所建立的如下关系：

$$p_1 - f_1 - rel(v) - f_2 - p_2. \quad (1)$$

其中 p_1 和 p_2 分别代表两个物体在状态 S 下，在参考坐标系中的位置； f_1 和 f_2 代表有关几何特征在各自物体局部坐标系中的位置； rel 是由关系 r 确定的一个函数； v 是由 rel 所确定的由 n 个变量构成的向量。几何推理系统交替地使用合并规则、串接规则和节点吸收规则等推理规则对网络中的关系进行综合，以生成限制性更强的新关系。

合并规则是在两个物体位置之间存在关系对



时使用的。推理系统在合并规则表中寻找适当的入口以便在 p_1 和 p_2 之间生成新的关系

$$p_1 - f_{13} - rel_3 - f_{23} - p_2,$$

以代替原来的关系对。

串接规则是在三个物体之间存在有关系链

$$p_1 - f_{11} - rel_1 - f_{21} - p_2 - f_{22} - rel_2 - f_{31} - p_3$$

时使用的。推理系统在串接规则表中寻找适当的入口以便在物体位置 p_1 和 p_3 之间生成一新关系

$$p_1 - f_{12} - rel_3 - f_{32} - p_3.$$

节点吸收规则用以检查在每一形如(1)的关系中，两个物体之间的相对位置是否可以完全确定。如果可以的话，则将其中一个物体位置节点（例如 p_2 ）吸收进另一个物体位置节点（例如 p_1 ），并将所有包含有 p_2 的关系转换成包含 p_1 的关系。

由于交替使用这些规则，关系网络中关系的数目是单调递减的。推理过程一直进行到网络中没有合适的关系可以被综合为止。这时每一环境状态或者是完全确定的，或者是由于用户程序提供的信息不够而仍有某些不确定之处。

三、验证视觉系统和 RAPT 中的视觉命令

有时由于某些设备的精度不够或是其它原因，某些实际环境状态与用户程序的描述有所不同，因此有些物体的实际位置与推导出的预期位置也会有差异。这些差异有可能导致机器人任务的失败。为此，用户可以使用视觉信息来探测物体的实际位置并相应地调整机器人的动作。这种提供视觉信息的系统称为验证视觉系统^[3]。验证视觉系统的主要任务是检查并确定物体的实际位置而不是识别物体。这种视觉系统很适于在机器人系统中使用。一方面，机器人的工作环境不是杂乱无章的随机环境，且被操作的物体的种类、形状及其大体位置通常都是已知的。不确定性只体现在某些物体的实际位置上。另一方面，机器人系统，特别是使用比较高级语言的系统，可以向视觉系统提供其所必须的信息，如物体的预期位置及模型等等，以控制视觉系统的工作和解释视觉信息。

在 RAPT 中，视觉系统提供以下主要命令：

1) LOOK. 这条命令指定需要视觉系统检验的物体的几何特征以及所使用的摄像机。

2) TOLERANCE. 指明被检验物体实际位置的最大偏移范围。这个信息可供视觉系统确定在执行 LOOK 命令时的搜索范围。

3) INVIOLEATE. 告诉视觉系统有关被检验物体的实际位置的某些限制。这些限制有助于正确地解释视觉信息。

下面是一组视觉命令的例子，其中 COMBINE, TERCOM 是视觉命令括号。

COMBINE:

LOOK/edgel of body1, CameraA;

LOOK/edge2 of body1, CameraB;

INVIOLEATE/against, bottom of body1, top of table;

TERCOM;

这一组命令告诉视觉系统使用 Camera A 和 camera B 分别观察物体 body 1 的两条边，并在解释视觉信息时要考虑到 body 1 是放在工作台上，即 Body 1 的 bottom 与 table 的 top 保持关系 against 这一限制。用户可以在其程序的任何地方插入一组适当的视觉命令，以对实际位置有可能变动的物体进行检验。在 RAPT 中，摄像机在位置和运动方面也是作为普通物体处理的，因此其位置也可以用其它摄像机来检验。这在使用安装于运动部件，如机器人手部装置上的摄像机时有其重要意义。

四、视觉信息的基本处理方式

RAPT 在处理视觉信息时工作方式如下：

首先，编译系统对视觉信息进行符号性解释，或称为符号推理。这是由于为了能在执

行阶段实时地利用视觉信息，需要在脱机编译阶段对视觉信息进行预处理。而由于实际的视觉信息要在执行阶段才能由摄像机获取，这种在编译阶段进行的预处理只能符号式地进行。符号推理的结果是对每一被视觉系统检验的物体的位置产生一个包含有若干变量的符号表达式。当用户程序实际运行时，这些变量被相应的视觉信息代真后，即可计算出符号表达式的值，从而确定了被检验的物体的实际位置。

然后，编译系统进一步分析视觉信息对各个物体在每一环境状态下的位置信息以及机器人的预期动作所产生的影响，并将这些结果保存在一个框架结构中以便后置处理模块使用。在机器人的工作过程中，每一环境状态都不是孤立的而是相互影响的。同样，在每一环境中不同物体的位置也是相互影响的。例如，假设物体 A 与物体 B 被装配在一起。如果视觉验证发现物体 A 的实际位置偏离了预期值，那么可以肯定物体 B 的实际位置也必然偏离了其预期位置。因此不仅涉及到物体 A 的操作要做修正，涉及到物体 B 的操作也应该进行修正。通过框架结构的建立，编译系统使得视觉信息可以被最大限度地利用，以便机器人系统克服工作环境中不确定性的影响。

符号推理的工作是由符号推理系统来完成的。限于篇幅，符号推理系统的工作原理及其实现将另文讨论。下面将着重讨论框架结构的建立及其规则。

五、物体位置与视觉信息的关系

物体位置根据其与其与视觉信息的关系可以分为三类：第一类是直接被视觉系统所检验的位置；第二类是与视觉信息无关的位置；第三类是与被视觉系统检验的物体位置保持一定的相互关系，因而与视觉信息间接有关的位置。为了在编译阶段统一处理这三种情况，引入了修正因子的概念。用户程序中每个物体在每一状态下的位置都对应一个修正因子，这个修正因子指明该物体预期位置与实际位置的差异。假设物体 b 在状态 i 中的预期位置矩阵是 PN_{bi} ，实际位置矩阵是 PA_{bi} ，修正因子是 FM_{bi} ，* 表示矩阵乘法，则有

$$FM_{bi} * PN_{bi} = PA_{bi}. \quad (2)$$

对于直接被视觉系统检验的位置，其修正因子为

$$FM_{bi} = PA_{bi} * PN_{bi}^{-1}.$$

这里 PA_{bi} 是由视觉信息确定的该物体的实际位置。在编译阶段它只是一个符号表达式，所以相应的 FM_{bi} 也是一个符号表达式。对于与视觉信息无关的位置，其修正因子为单位矩阵。对于与视觉信息间接有关的位置，其修正因子与相关位置的修正因子有关，其具体表达式则由有关位置间的相互联系所决定。下面讨论两种简单的情况。

第一种情况，假设物体 b 在状态 $(i+1)$ 时的位置与其在状态 i 时的位置保持一个预定的关系 T_{bi} ，即在预期状态中有

$$PN_{b(i+1)} = PN_{bi} * T_{bi}. \quad (3)$$

而在实际状态中有

$$PA_{b(i+1)} = PA_{bi} * T_{bi}. \quad (4)$$

这相当于 RAPT 中的目标非确定性动作。由(2),(3),(4)式可知

$$PA_{b(i+1)} = PA_{bi} * T_{bi} = FM_{bi} * PN_{bi} * T_{bi}$$

$$= FM_{bi} * PN_{b(i+1)} = FM_{b(i+1)} * PN_{b(i+1)},$$

即有

$$FM_{b(i+1)} = FM_{bi}. \quad (5)$$

第二种情况,假设物体 a 与物体 b 在状态 i 中保持一预定的相互关系 RP_{ab} , 即

$$RP_{ab} = PN_{ai} * PN_{bi}^{-1} = PA_{ai} * PA_{bi}^{-1}, \quad (6)$$

这相当于 RAPT 中目标确定性动作。由 (2), (6) 式可知

$$\begin{aligned} PA_{ai} &= PN_{ai} * PN_{bi}^{-1} * PA_{bi} = PN_{ai} * PN_{bi}^{-1} * FM_{bi} * PN_{bi} \\ &= PN_{ai} * PN_{bi}^{-1} * FM_{bi} * PN_{bi} * PN_{ai}^{-1} * PN_{ai} = FM_{ai} * PN_{ai}. \end{aligned}$$

即有

$$FM_{ai} = PN_{ai} * PN_{bi}^{-1} * FM_{bi} * PN_{bi} * PN_{ai}^{-1}. \quad (7)$$

六、视觉信息框架结构的建立规则

视觉信息框架结构的主要功能是有效地表示物体位置与视觉信息之间的关系,其核心是一个修正因子数组。每个物体在每一环境状态下的位置都对应着这个数组中的一个元素。数组元素的内容根据其所对应的物体位置与视觉信息的关系可能有三种不同的形式。如果一个物体在某一个环境状态下被视觉系统直接检验,则其所对应的数组元素是一个包含视觉信息变量的符号表达式。如果一个物体在某一环境状态下的位置与视觉信息无关,则其所对应的数组元素是一个单位矩阵 I 。如果一个物体在某一环境状态下的位置与视觉信息间接有关,则其所对应的数组元素是一个指针或指针三元组,这取决于该物体位置与视觉信息之间的关系类型。

修正因子数组元素的内容是按如下步骤确定的。首先,每个物体在起始状态下的修正因子数组元素都是单位矩阵 I 。这是因为在起始状态中视觉系统还没有开始工作,因此假定起始状态的实际情况与用户程序的描述一致。在此后的各种状态下,每个将被视觉系统直接检验的物体位置的数组元素都设置成相应的符号表达式,而其它物体所对应的数组元素则根据指针建立规则设置为不同类型的指针或指针三元组。这些指针根据物体的运动情况及相互关系指向修正因子数组中同一状态下或以前状态下的数组元素。例如,对经过目标非确定性运动的简单物体来说,(5)式所表示的修正因子之间的关系,可以用从 $FM_{b(i+1)}$ 指向 FM_{bi} 的水平指针表示,而对经过目标确定性运动的简单物体,(7)式所表示的关系可以用从 FM_{ai} 指向 FM_{bi} 的垂直指针来表示。处于联结或部件中的物体的修正因子间的复杂关系则需要用指针三元组表示^[4]。对于与视觉信息无关的物体位置来说,它所对应的指针(包括三元组中的指针)是通过指针链指向起始状态中的单位矩阵的。对于与视觉信息间接有关的物体位置,它所对应的指针则通过指针链指向相关的符号表达式。在所有的元素都被设置完毕后,修正因子数组根据一组简化规则进行简化。所有指向单位矩阵的指针都被替换成为单位矩阵,而指向符号表达式的指针链则在一定的条件下被缩短。满足一定条件的指针三元组也被替换成单个的指针。简化工作可以减少实时视觉信息处理和使用时的计算量。

指针建立规则分为四组,分别适用于简单物体,联结和部件中的物体以及特殊状态下

的摄像机。指针建立规则的详细讨论可见文献 [4]。图 3 是根据指针建立规则对下面的 RAPT 程序段所设置的修正因子数组的内容(未简化):

```

remark bodies b1 b2 b3 have been defined;
.....
verify/b2;    remark abbreviation for vision command
              package, now in situation i;
move/b1;
  fixed/b1, b2; remark abbreviation for a set of relations which completely defines the position of b1 with
              respect to b2, sit i + 1;
tied/b1, b2;
verify/b3;    remark sit i + 2;
move/b2;
  fixed/b2, b3;    remark sit i + 3;
.....
  
```

修正因子数组确定了视觉信息与物体实际位置之间的关系。在程序执行阶段,一旦得到了相应的视觉信息,每个相关修正因子的实际数值即可迅速计算出来,所有有关物体的实际位置就可以实时地确定,机器人控制系统也就可以根据物体的实际位置实时地调整其动作,完成预期的任务。在处理视觉信息的框架结构中,把根据用户程序得出的物体的预期位置与根据视觉信息得到的对预期位置的修正值分开表示和处理,使得编译系统的几何推理工作不受引入的传感信息类型和处理方法的影响,同时也给后置处理和机器人的动作修正方式带来很大的灵活性,使得对机器人的控制策略可以适应不同的情况和要求。限于篇幅,兹不赘述¹⁾。图 4 是包含视觉控制功能的 RAPT 系统的数据流图。

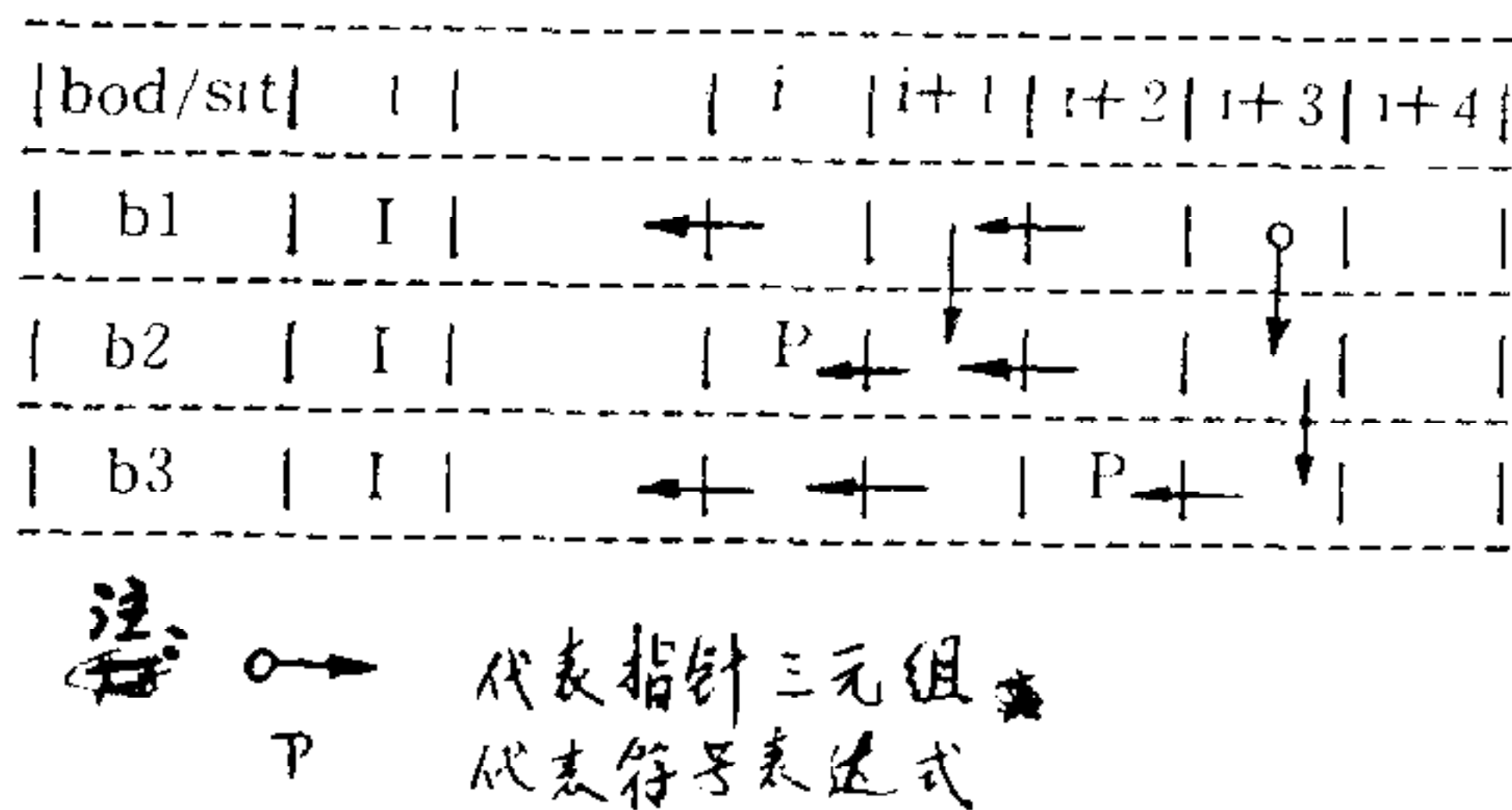


图 3 一个 RAPT 程序段及相应的修正因子数组

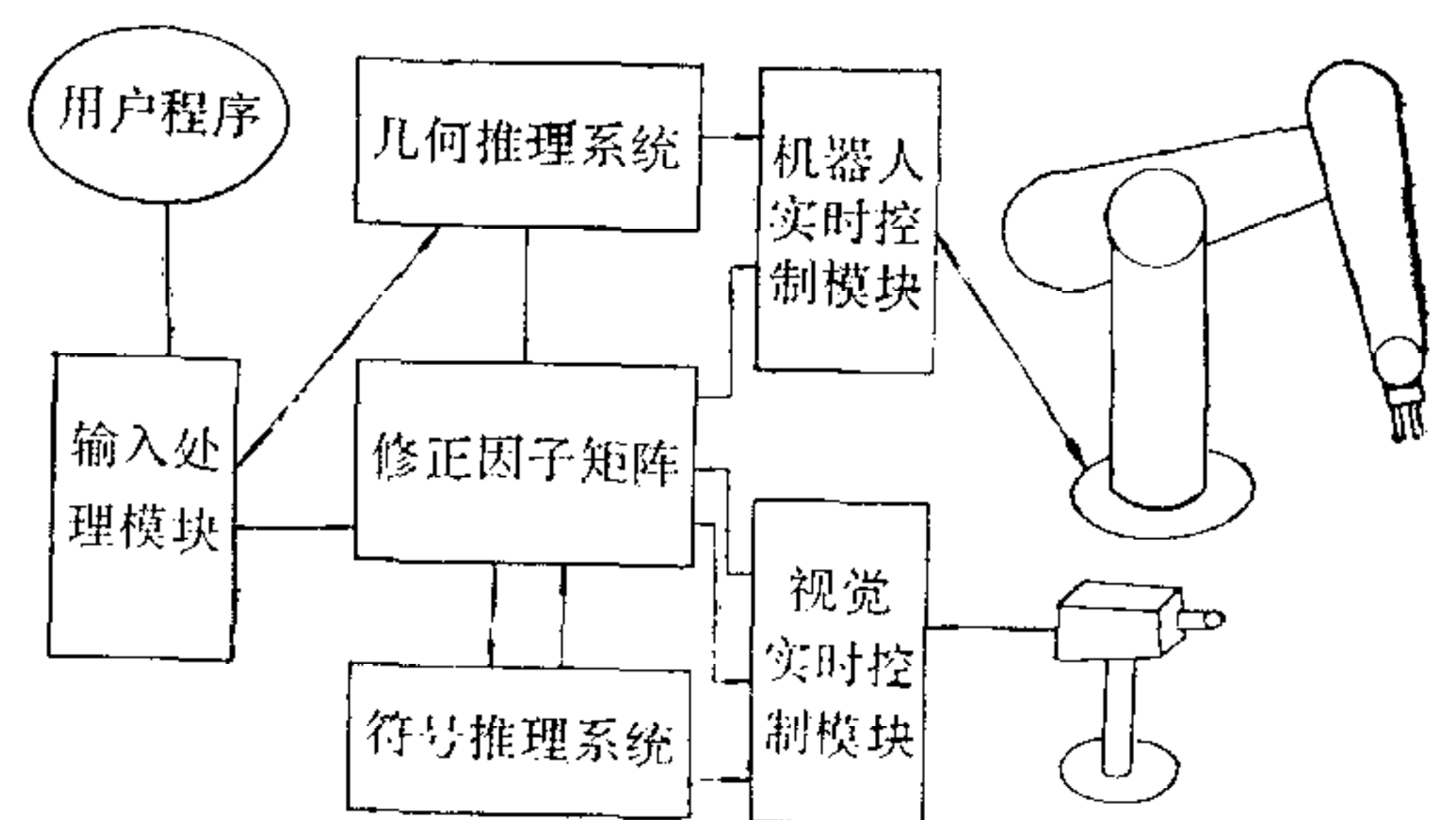


图 4 带视觉控制功能的 RAPT 系统的数据流图

结 语

本文讨论了一种在物体级机器人语言中使用视觉信息的方法。这种方法在脱机编译阶段对视觉信息进行了大量的符号处理,因此使得机器人程序在执行阶段可以实时地使用视觉信息来修正机器人的动作。由于引入了视觉信息框架,系统可以有效地利用视觉信息,以更新其有关工作环境状态的信息。除了视觉信息外,这种方法还可以用于处理其它类型的传感信息。

1) 尹宝林,北京航空学院科研报告 BH-B1558, 1984.

参 考 文 献

- { 1] Popplestone, R. J., Ambler, A. P., A language for Specifying Robot Manipulations in Robot Technology, *IEE Control Engineering* 23(1983).
- { 2] Corner, D. et al., Reasoning About the Spatial Relationships Derived from a RAPT Program for Describing Assembly by Robot, Proc. of IJCAI-83, p. 842.
- { 3] Bolles, R. C., Verification Vision for Programmable assembly, Proc. of IJCAI-77, p. 563.
- { 4] Yin Baolin, Combining Vision Verification with a High Level Robot Programming Language, Ph. D. Thesis, Univ. of Edinburgh, UK 1984.

AN IMPLEMENTATION METHOD OF THE VISION FACILITIES IN A HIGH LEVEL ROBOT PROGRAMMING LANGUAGE

YIN BAOLIN

(Beijing Institute of Aeromautics and Astronautics)

ABSTRACT

In this paper, a method which uses the vision information in a high level robot programming language RAPT is described. The vision information is used to determine the difference between the planned positions and the actual positions of objects in the robot's environment, and adjust the robot's actions accordingly. By using symbolic reasoning, the main calculation for vision information is done at the compile time so that the run time vision task can be performed at real time. A robust framework for handling vision data is provided, so that all ramifications of the effect of the vision information are taken into account.