

一种从轮廓恢复三维形状的方法

程义民 孙启彬 王 敏

(中国科技大学电子技术部,合肥)

摘 要

本文描述了一种基于表面方向及恢复树,从景物轮廓线求三维景物表面形状的方法。文中还描述了两个消影点,一个消影点及无消影点时,计算表面方向的新算法,并给出一个简单例子。

关键词: 恢复树,透视投影,四边形,表面方向。

一、引 言

从二维图象求得景物的三维信息是计算机视觉、景物分析、机器人视觉等研究领域的重要课题^[1,2]。通常先求景物的2.5维信息,如深度、表面方向^[3]等,进而求得景物的三维结构。2.5维信息常可由双眼或多眼图象^[4]、运动图象^[5]、纹理^[6]等求得,并已有了一些应用这些方法的软件及硬件系统^[7,8]。

从成像系统获得的图象^[9],经过处理,可得到景物轮廓线^[10]。本文描述了一种基于表面方向及恢复树,从景物轮廓线求三维景物表面形状的方法。该方法主要是由景物轮廓线求得表面方向,再从图的表示求得恢复树,最后由恢复树求出三维景物的表面形状。它可用于机器人视觉系统中,某些机器零部件及室内景物的三维识别及理解。

二、表面方向的计算

计算机视觉问世以来,三维景物表面方向的计算得到广泛的研究,已有许多关于表面方向的计算方法^[1,2],本文仅讨论摄象机一阶近似时,基于射影几何的几种新算法。

在一阶近似条件下,二维图象是三维景物在图象平面上的透视投影^[1]。表面为平面(或小平面)的物体,若其表面可由两组共面平行线或平行四边形确定,如机器零部件某些面的边缘、建筑物表面上的窗户等,则该空间平行四边形在图象平面上的投影,通常仍为四边形。其两对边可能都不平行或仅有一对边平行,如图1,这样可求得二个消影点或只能求得一个消影点。但不管哪种情况都可求得表面方向,且空间平面上只有一组平行线段或点列时,也可求得表面方向。

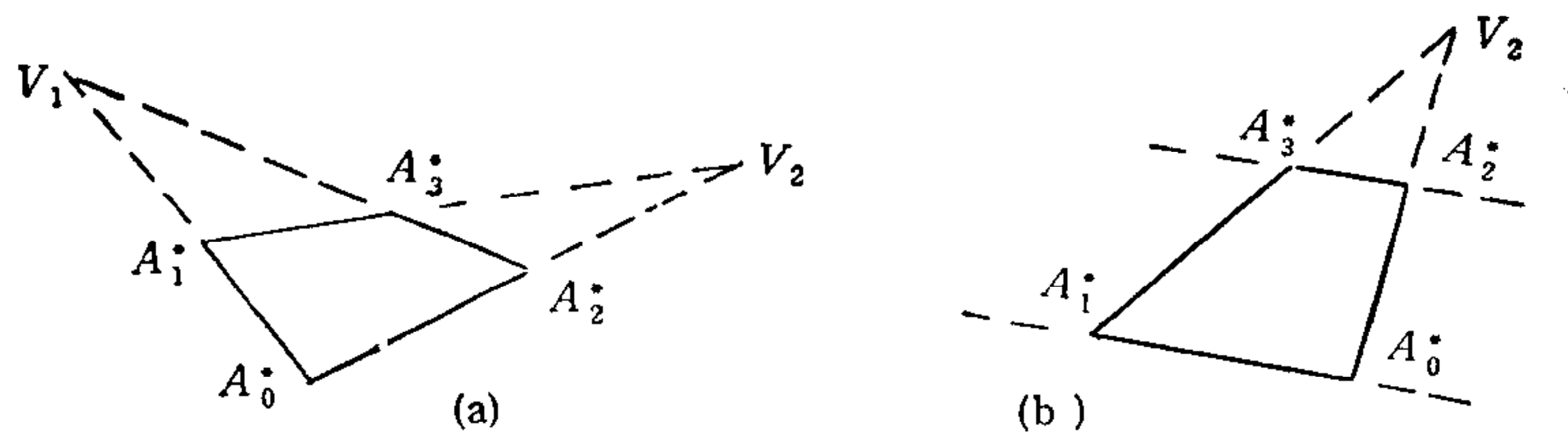


图 1 平行四边形在图象平面上的透视投影

在三维坐标,任一空间平面 A 可表示为

$$z = px + qy + D. \tag{1}$$

其法矢为 $n = (p, q, -1)$, 即 (p, q) 空间的一点. 取图象平面 A^* 和 xoy 平面重合, 透视投影中心为 $S(0, 0, -f)$, 其中 f 为摄象机的焦距. 任一空间点 (x, y, z) 到平面 A^* 的透视变换为^[11]

$$\begin{aligned} x^* &= x / (rz + 1), \\ r^* &= y / (rz + 1). \end{aligned} \tag{2}$$

其中 $r = 1/f, (x^*, y^*)$ 是图象平面坐标. 可以求得消影线方程为^[3]

$$px^* - qy^* - f = 0. \tag{3}$$

若已知两个消影点, 由方程(3)可得平面 A 的表面方向 (p, q) .

2.1 由两个消影点求表面方向

若一空间平面 A 上的平行四边形, 在图象平面上的透视投影为四边形 $A_0^*A_1^*A_2^*A_3^*$, 两对边不互相平行, 如图 1(a), 则两对边 $A_0^*A_1^*, A_2^*A_3^*$ 所在直线 L_0^*, L_1^* 的直线方程可记为

$$\begin{aligned} a_0x^* + b_0y^* + c_0 &= 0, \\ a_1x^* + b_1y^* + c_1 &= 0. \end{aligned} \tag{4}$$

其中 $a_0 = y_1^* - y_0^*, b_0 = x_0^* - x_1^*, c_0 = x_1^*y_0^* - x_0^*y_1^*, a_1 = y_3^* - y_2^*, b_1 = x_2^* - x_3^*, c_1 = x_3^*y_2^* - x_2^*y_3^*$.

解方程组(4), 可得消影点 V_1 的坐标

$$v_{1x} = \frac{\begin{vmatrix} b_0 & c_0 \\ b_1 & c_1 \end{vmatrix}}{\begin{vmatrix} a_0 & b_0 \\ a_1 & b_1 \end{vmatrix}}, \quad v_{1y} = -\frac{\begin{vmatrix} a_0 & c_0 \\ a_1 & c_1 \end{vmatrix}}{\begin{vmatrix} a_0 & b_0 \\ a_1 & b_1 \end{vmatrix}}, \tag{5}$$

$$a_0b_1 \neq a_1b_0.$$

由于 A^* 和 xoy 平面重合, V_1 在三维坐标系中, 坐标为 $(v_{1x}, v_{1y}, 0)$. 若令 n_{V_1} 为透视投影中心 $S(0, 0, -f)$ 指向 V_1 的矢量, 则

$$n_{V_1} = (v_{1x}, v_{1y}, f).$$

它和平面 A 法矢 n 的点积为

$$n_{V_1} \cdot n = (v_{1x}, v_{1y}, f) \begin{pmatrix} p \\ q \\ -1 \end{pmatrix} = pv_{1x} + qv_{1y} - f. \tag{6}$$

又由于 V_1 为消影点, 必满足消影线方程(3)

$$pv_{1x} + qv_{1y} - f = 0. \quad (7)$$

即 n_{V_1} 和 n 垂直, 直线 SV_1 和空间平面 A 平行.

同样, 由图象平面 A^* 上直线 $L_2^*(A_0^*A_2^*), L_3^*(A_1^*A_3^*)$, 可求得另一约束方程

$$pv_{2x} + qv_{2y} - f = 0. \quad (8)$$

方程(7),(8)对 p, q 求解, 可得

$$p = \frac{\begin{vmatrix} f & v_{1y} \\ f & v_{2y} \end{vmatrix}}{\begin{vmatrix} v_{1x} & v_{1y} \\ v_{2x} & v_{2y} \end{vmatrix}} = f \frac{\begin{vmatrix} a_0 & c_0 & a_2 & b_2 \\ a_1 & c_1 & a_3 & b_3 \end{vmatrix} - \begin{vmatrix} a_2 & c_2 & a_0 & b_0 \\ a_3 & c_3 & a_1 & b_1 \end{vmatrix}}{\begin{vmatrix} a_0 & c_0 & b_2 & c_2 \\ a_1 & c_1 & b_3 & c_3 \end{vmatrix} - \begin{vmatrix} a_2 & c_2 & b_0 & c_0 \\ a_3 & c_3 & b_1 & c_1 \end{vmatrix}}, \quad (9)$$

$$q = \frac{\begin{vmatrix} v_{1x} & f \\ v_{2x} & f \end{vmatrix}}{\begin{vmatrix} v_{1x} & v_{1y} \\ v_{2x} & v_{2y} \end{vmatrix}} = f \frac{\begin{vmatrix} a_2 & b_2 & b_0 & c_0 \\ a_3 & b_3 & b_1 & c_1 \end{vmatrix} - \begin{vmatrix} a_0 & b_0 & b_2 & c_2 \\ a_1 & b_1 & b_3 & c_3 \end{vmatrix}}{\begin{vmatrix} a_0 & c_0 & b_2 & c_2 \\ a_1 & c_1 & b_3 & c_3 \end{vmatrix} - \begin{vmatrix} a_2 & c_2 & b_0 & c_0 \\ a_3 & c_3 & b_1 & c_1 \end{vmatrix}}. \quad (10)$$

其中 $v_{1x}v_{2y} \neq v_{2x}v_{1y}$,

$$a_2 = y_2^* - y_0^*, \quad b_2 = x_0^* - x_2^*, \quad c_2 = y_0^*x_2^* - y_2^*x_0^*,$$

$$a_3 = y_3^* - y_1^*, \quad b_3 = x_1^* - x_3^*, \quad c_3 = y_1^*x_3^* - y_3^*x_1^*.$$

2.2 由一个消影点求表面方向

如果平行四边形在图象平面上的透视投影有一对边互相平行, 见图 1(b), 只可求得一个消影点, 如 $V_2(v_{2x}, v_{2y})$, 得一约束方程

$$pv_{2x} + qv_{2y} - f = 0. \quad (11)$$

为了求得 (p, q) 仍需另一约束方程.

若直线 $L_0^*(A_0^*A_1^*), L_1^*(A_2^*A_3^*)$ 在 A^* 上不相交, 则和消影线平行^[3], L_0^* 或 L_1^* 可表示为

$$px^* + qy^* - R = 0. \quad (12)$$

其法线方向为 $n_p = (p, q)$, 若 $n_d = (n_x, n_y)$ 表示它的方向矢量, 则有 $n_p \perp n_d$, 即

$$n_p \cdot n_d = pn_x + qn_y = 0. \quad (13)$$

这样, 可得另一约束方程, 方向矢量 n_d 可由图象平面 A^* 上的直线 L_0^* 或 L_1^* 求出.

式(13)和式(11)对 p, q 求解, 可得空间平面 A 的表面方向^[3].

2.3 从一组共面平行线的透视投影求表面方向

若空间平面 A 上仅有一组平行线段或点列, 在图象平面 A^* 上可得两类图象. 一类是图象平面上的透视投影不互相平行, 见图 2(a), 可求得一有限消影点 v_1 ; 另一类在图象平面仍为平行线, 不能得到消影点. 由前面讨论知, 这两种情况可各得一约束方程式(7)及式(13).

为了求另一约束方程, 可在图象平面 A^* 上任一直线 L^* , 和平行线组在 A^* 平面上的透视投影分别相交于 $A_0^*, A_1^*, A_2^*, A_3^*$, 见图 2(a)、(b), 考虑透视中心 S 和 $A_0^*A_3^*$ 所在平面 $SA_0^*A_3^*$, 见图 2(c), 该平面和 A 平面相交于直线 L , 则 A_0, A_1, A_2, A_3 是相应的空间景物点. 将平面 A 上这些点间的距离表示为

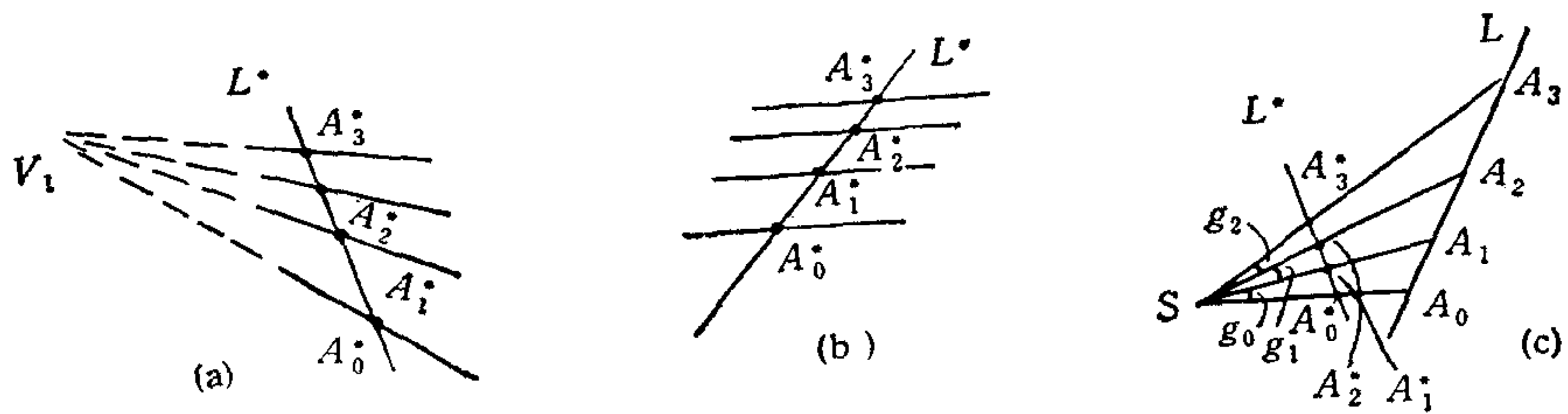


图2 一组共面平行线在图象平面的透视投影。(a) 有一消影点, (b) 无消影点, (c) $SA_0^*A_3^*$ 平面

$$a_0 = A_0A_1, \quad a_1 = A_1A_2, \dots, \quad a_{n-1} = A_{n-1}A_n. \quad (14)$$

在图象平面 A^* 上, 相应点的距离表示为

$$b_0 = A_0^*A_1^*, \quad b_1 = A_1^*A_2^*, \dots, \quad b_{n-1} = A_{n-1}^*A_n^*. \quad (15)$$

在 $SA_0^*A_3^*$ 平面上, 考虑三角形的面积, 则有

$$2\Delta SA_0^*A_2^* = SA_0^* \cdot SA_2^* \cdot \sin(g_0 + g_1) = (A_0^*A_1^* + A_1^*A_2^*) \cdot h^*. \quad (16)$$

h^* 是点 S 到直线 L^* 的距离。这样就有

$$A_0^*A_1^* + A_1^*A_2^* = (SA_0^* \cdot SA_2^* \cdot \sin(g_0 + g_1)) / h^*. \quad (17)$$

用同样的方法可求得

$$A_1^*A_2^* = SA_1^* \cdot SA_2^* \cdot \sin g_1 / h^*, \quad (18)$$

$$A_1^*A_2^* + A_2^*A_3^* = SA_1^* \cdot SA_3^* \cdot \sin(g_1 + g_2) / h^*, \quad (19)$$

$$A_0^*A_1^* + A_1^*A_2^* + A_2^*A_3^* = SA_0^* \cdot SA_3^* \cdot \sin(g_0 + g_1 + g_2) / h^*. \quad (20)$$

由式(17)、(18)、(19)、(20)可求得交比

$$\begin{aligned} R(A_0^*, A_1^*, A_2^*, A_3^*) &= \frac{(b_0 + b_1)(b_1 + b_2)}{b_1(b_0 + b_1 + b_2)} \\ &= \frac{\sin(g_0 + g_1) \cdot \sin(g_1 + g_2)}{\sin g_1 \cdot \sin(g_0 + g_1 + g_2)}. \end{aligned} \quad (21)$$

同样, 点列 A_0, A_1, A_2, A_3 的交比为

$$\begin{aligned} R(A_0, A_1, A_2, A_3) &= \frac{(a_0 + a_1)(a_1 + a_2)}{a_1(a_0 + a_1 + a_2)} \\ &= \frac{\sin(g_0 + g_1) \cdot \sin(g_1 + g_2)}{\sin g_1 \cdot \sin(g_0 + g_1 + g_2)}. \end{aligned} \quad (22)$$

由式(21)和(22)知

$$R(A_0^*, A_1^*, A_2^*, A_3^*) = R(A_0, A_1, A_2, A_3),$$

即

$$\frac{(b_0 + b_1)(b_1 + b_2)}{b_1(b_0 + b_1 + b_2)} = \frac{(a_0 + a_1)(a_1 + a_2)}{a_1(a_0 + a_1 + a_2)}.$$

若 A 上各平行线间的距离比已知, 则可求得 b_i 间的关系。为了简便, 假定平行线等距, 则

$$\begin{aligned} R(A_0, A_1, A_2, A_3) &= 4/3, \\ b_2 &= (b_1b_0 - b_1^2) / (3b_0 - b_1). \end{aligned} \quad (23)$$

写成迭代形式

$$b_n = (b_{n-1}b_{n-2} - b_{n-1}^2)/(3b_{n-2} - b_{n-1}). \quad (24)$$

显然, 当 $b_n \rightarrow 0$ 时, L^* 必和消影线相交.

这样就可以通过以下步骤求得另一消影点, 进而可求得表面方向.

1) 在图象平面 A^* 求 L^* 的直线方程

$$y^* = Kx^* + B. \quad (25)$$

其中 $K = (y_2^* - y_0^*)/(x_2^* - x_0^*)$, $x_2^* - x_0^* \neq 0$, $B = y_0^* - Kx_0^*$,

(x^*, y^*) 是图象平面 A^* 上的坐标(图 2).

2) 计算初始值

$$\begin{aligned} b_0 &= \sqrt{(x_1^* - x_0^*)^2 + (y_1^* - y_0^*)^2}, \\ b_1 &= \sqrt{(x_2^* - x_1^*)^2 + (y_2^* - y_1^*)^2}, \\ M &= b_0 + b_1. \end{aligned} \quad (26)$$

3) 计算

$$\begin{aligned} b_n &= (b_{n-1}b_{n-2} - b_{n-1}^2)/(3b_{n-2} - b_{n-1}), \\ M &= M + b_n. \end{aligned}$$

4) 如果 $b_n > \delta$, 再回到第三步.

5) 对 x^*, y^* 求解方程组

$$\begin{aligned} (x^* - x_0^*)^2 + [(Kx^* + B) - y_0^*]^2 &= M^2, \\ y^* &= Kx^* + B. \end{aligned} \quad (27)$$

解 (x^*, y^*) 为又一消影点, 记为 $V_2(v_{2x}, v_{2y})$.

6) 解方程组

$$\begin{aligned} pv_{1x} + qv_{1y} - f &= 0, \\ pv_{2x} + qv_{2y} - f &= 0, \\ v_{1x}v_{2y} &\neq v_{1y}v_{2x}. \end{aligned}$$

或

$$\begin{aligned} pn_x + qn_y &= 0, \\ pv_{2x} + qv_{2y} - f &= 0, \\ n_xv_{2y} &\neq n_yv_{2x}. \end{aligned}$$

即可求得空间平面 A 的表面方向.

在计算时, 改变直线 L^* 的斜率, 可改变迭代计算的收敛速度.

三、从表面方向求三维形状

求得物体表面方向后, 就可求景物的三维形状. 为了求得实际景物的三维坐标, 至少要知一个深度坐标 z . 但在许多计算机视觉及图象理解任务中, 如三维零部件的识别等, 通常只需求出景物的相对坐标及形状. 该形状和实际尺寸仅差一比例系数. 下述方法可用于凸多面体类景物表面形状的恢复.

3.1. 图表示及恢复树

机器部件及某些景物的图象,经过预处理可得类似线条画的轮廓线,见图 3(a),再经特征抽取、分类,可将有关的点、线及相关的表面及表面方向归为一类.这时轮廓线的几何结构及 2.5 维信息可用图来表示.图 3(b) 给出了图 3(a) 的图表示.图的顶点表示物体的表面及相关信息,相邻两表面之间有一条边相连.

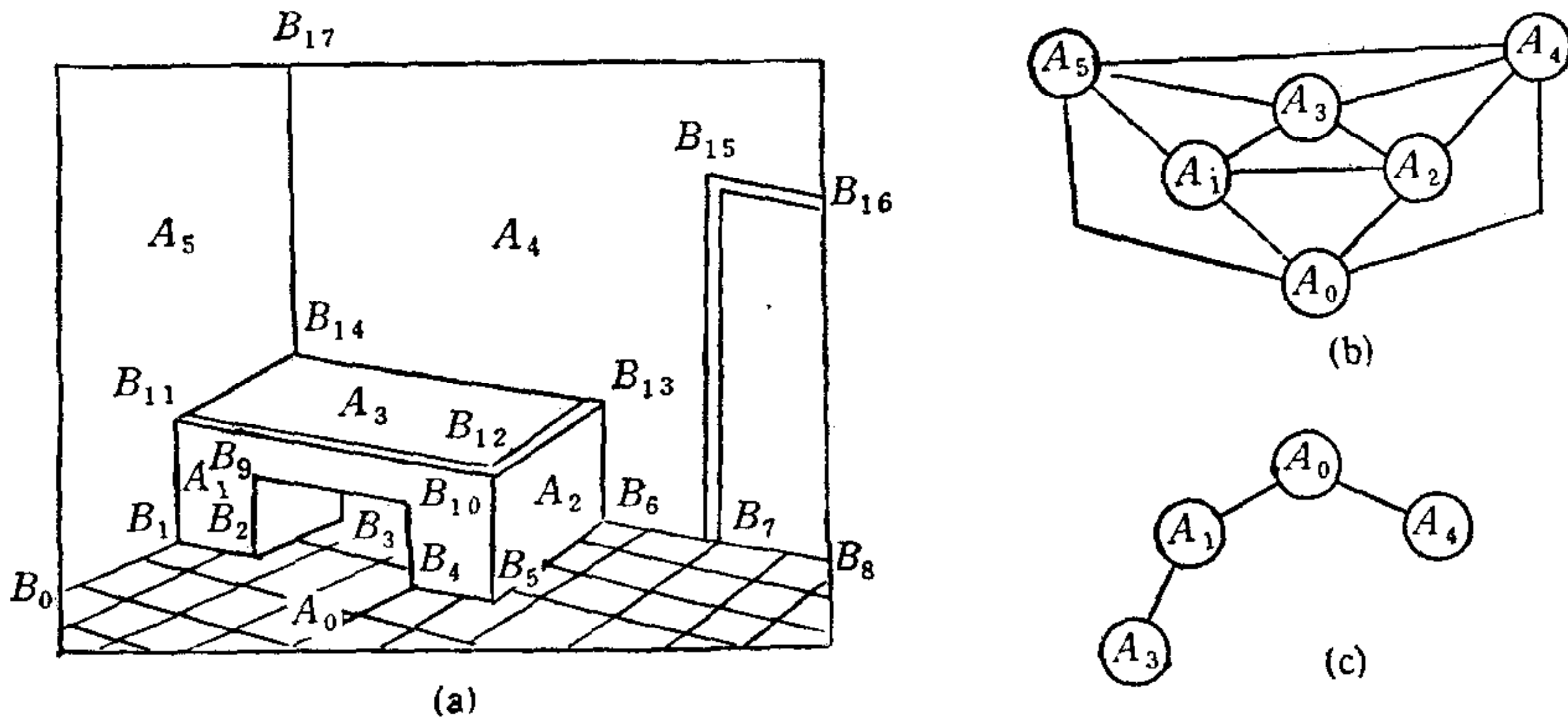


图 3 某景物图象

(a) 室内景物; (b) 景物的图表示; (c) 恢复树.

为了求恢复树,可把图顶点 A_i 表示为

$$A_i = (p_i, q_i, B_0, B_1, \dots, B_n, \dots, B_l B_{l+1} \dots B_m)$$

其中, p_i, q_i 是平面 A_i 的表面方向; $B_0 B_1 \dots B_n, \dots, B_l B_{l+1} \dots B_m$ 是 A_i 边界线上相邻关键点序列.图中相邻顶点间至少有一个相同关键点(通常有两个).这样,若以包含景物所有关键点为条件,对图搜索,可得一包含最少顶点的树,图 3(c) 给出 (b) 的一个树.从该树可恢复景物的三维形状,该树可称为恢复树.

图 3(c) 中各顶点为

$$A_0 = (p_0, q_0, B_0 B_1 B_2 B_3, B_4 B_5 B_6 B_7 B_8), \quad A_1 = (p_1, q_1, B_1 B_2 B_9 B_{10} B_4 B_5 B_{12} B_{11} B_1),$$

$$A_3 = (p_3, q_3, B_{11} B_{12} B_{13} B_{14} B_{11}), \quad A_4 = (p_4, q_4, B_{17} B_{14} B_{13} B_6 B_7 B_{15} B_{16}).$$

从图 3(c) 知,恢复树只需包含所有关键点,不必包含所有顶点,即只需求出恢复树上的顶点,不必求出景物所有面的表面方向.当求得恢复树后,即可恢复景物的三维表面形状.在树根中选一关键点,如 A_0 中选 B_0 为参考点,且令 $B_0 = (x_0, y_0, z_0)$. 这时就有

$$x_0 = x_0^*(r z_0 + 1), \tag{28}$$

$$y_0 = y_0^*(r z_0 + 1),$$

$$D_0 = z_0 - p_0 x_0 - q_0 y_0. \tag{29}$$

平面 A_0 上的任一关键点 B_i , 满足如下等式:

$$p_0 x_i + q_0 y_i + D_0 = z_i,$$

$$x_i = x_i^*(r z_i + 1), \tag{30}$$

$$y_i = y_i^*(r z_i + 1).$$

对 x_i, y_i, z_i 解方程组(30),可求出 B_i 的三维空间坐标.由于树上任相连二顶点上,至少有一个公共关键点,可由 A_i 的信息求出相邻后续顶点的参数 D_i , 如 B_4 (图 3). 由式(30)可得 B_4 的空间坐标 (x_4, y_4, z_4) . 用类似的方法

$$D_i = z_i - p_i x_i - q_i y_i,$$

可求出 D_1 。这时对 A_1 上的任一关键点 $B_i(x_i, y_i, z_i)$ 求解方程组

$$p_1 x_i + q_1 y_i + D_1 = z_i,$$

$$x_i = x_i^*(r z_i + 1),$$

$$y_i = y_i^*(r z_i + 1).$$

可求出 A_1 上任一点 B_i 的三维坐标。以此类推, 遍历恢复树, 则可求出以关键点为基础, 包括它们间拓扑关系的景物表面三维形状表示。

3.2. 简例

我们已在微型计算机上实现了上述恢复过程, 并得到有意义的结果。下面给出一个简例及计算机计算结果。

图 4 是一个房子的轮廓线, 该图是由计算机产生的, 其关键点坐标为

$$A_0 = (-5, -153), \quad A_1 = (37, -222), \quad A_2 = (109, -200),$$

$$A_3 = (109, -160), \quad A_4 = (140, -152), \quad A_5 = (140, -191),$$

$$A_6 = (217, -167), \quad A_7 = (224, -102), \quad A_8 = (42, -137),$$

$$A_9 = (37, -133), \quad A_{10} = (12, -72), \quad A_{11} = (-7, -90),$$

$$A_{12} = (171, -57), \quad A_{13} = (-5, -92).$$

用上述方法求得 B_0, B_1, B_2 的表面方向为

$$B_0 = (-2.7474, 0.0000, -1.0000),$$

$$B_1 = (0.3640, 0.0000, -1.0000),$$

$$B_2 = (0.3640, 4.4562, -1.0000).$$

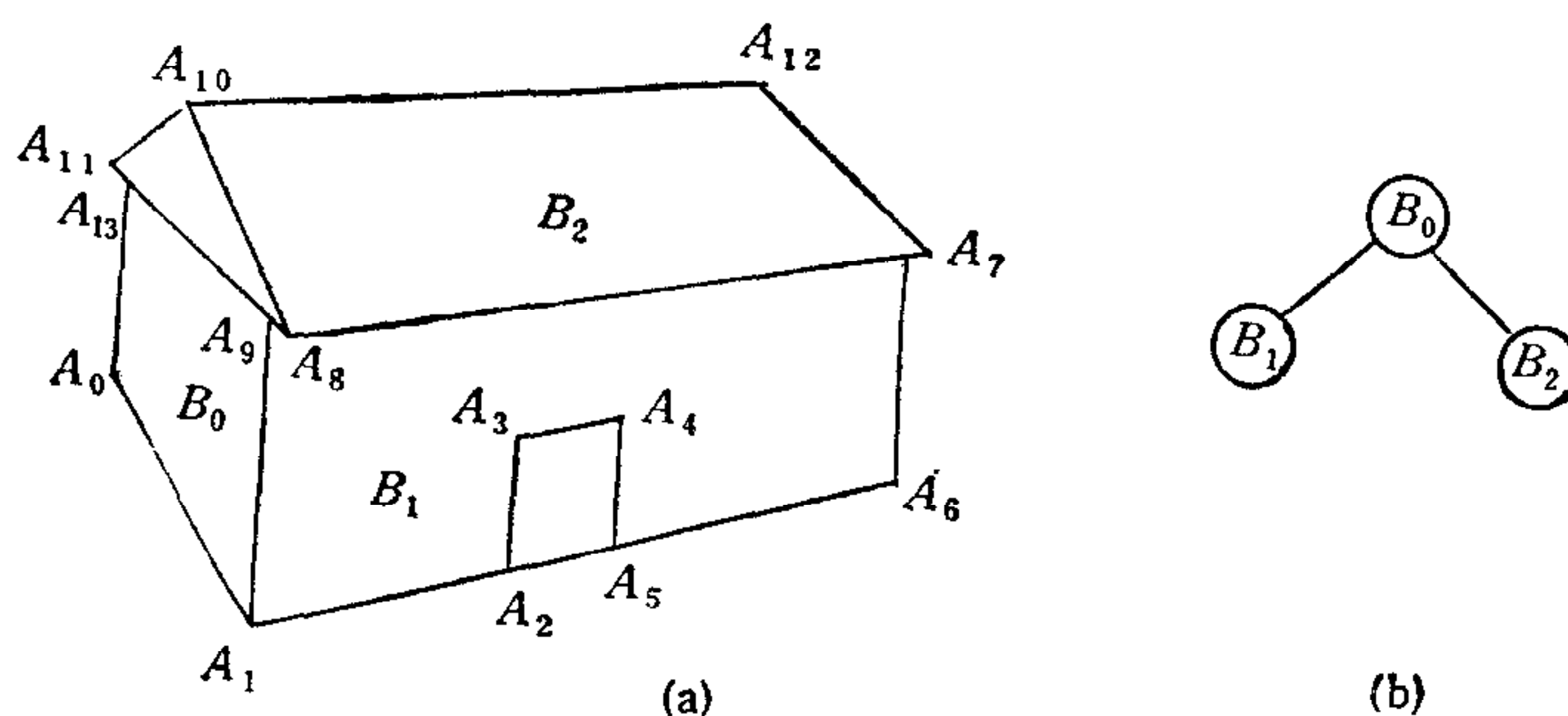


图 4 房子的轮廓(a)及恢复树(b)

其关键点的三维空间坐标, 由计算机求得为

$$A_0 = (-17.26, -499.75, 631.77), \quad A_1 = (85.29, -499.75, 350.00),$$

$$A_2 = (273.13, -499.75, 418.37), \quad A_3 = (273.13, -399.80, 418.37),$$

$$A_4 = (367.05, -399.80, 452.55), \quad A_5 = (367.05, -499.75, 452.55),$$

$$A_6 = (648.82, -499.75, 555.11), \quad A_7 = (648.82, -299.85, 535.21),$$

$$A_8 = (92.12, -299.85, 331.22), \quad A_9 = (85.29, -299.85, 350.00),$$

$$A_{10} = (34.01, -199.90, 490.88), \quad A_{11} = (-24.10, -299.85, 650.55),$$

$$A_{12} = (597.55, -199.90, 695.99), \quad A_{13} = (-17.26, -299.85, 631.76),$$

$$A_{14} = (546.27, -299.85, 836.87), \quad A_{15} = (648.82, -299.85, 555.11),$$

$$A_{16} = (546.27, -499.75, 836.87), \quad A_{17} = (539.43, -299.85, 855.66).$$

其中参考点选 $z_1 = 350$, 不可见关键点 $A_{14}, A_{15}, A_{16}, A_{17}$ 可由景物的对称性求得。

四、结 束 语

本文描述了一种从轮廓线求景物三维表面形状的方法,并给出了一个简例。结果表明该方法是可行的。

该方法主要针对凸多面体类景物,如某些机器零部件,家俱等室内景物及建筑物等人造物体表面形状的恢复,也可推广到表面为某些数学面的景物形状恢复。进而可用于三维识别、视觉指导下的机器人装配,自走机器人室内路径规划,及某些室外景物的三维理解及分析。

参 考 文 献

- [1] Ballard, D. H. Brown, C. M., "Computer Vision" Prentice-Hall, Englewood Cliffs, New Sersey, 1982.
- [2] Horn B. K. P. Robot Vision, The MIT Press, Cambrige, Massachusetts, 1986.
- [3] 程义民,孙启彬,表面方向的计算,中国科学技术大学学报,19(1989),2,223.
- [4] Marr, D., Poggio, T., Cooperative Computation of Disparity, Science 194, 1976, 283.
- [5] Prazdny, K., On the Information in Optical Flows, Computer Vision, Graphics and Image Processing, 22 (1983), 239.
- [6] Kender, J. R., Shape from Texture: A Brif Overview and A New Aggregation Transform, Proc. DARPAIV workshop, November 1978, 79.
- [7] Herman, M., Kanade, T., Incremented Reconstruction of 3-D Scenes from Multiple Image, Artificial Intelligence, 30(1986), 3, 289.
- [8] Wang Xufa, Cheng Yimin, Sun Qibin, Chen Xianfu, A Robot Vision System Controlled by Microcomputer, Proc. of the IEEE AEC-87, Hong Kong, Sept. 1987, 475.
- [9] 程义民,一种数字图象信号快速获取方法,自动化学报,4(1988),1,51.
- [10] Rosenfeld, A., Kak, A. C., Digital Picture Processing, Academic Press, 1976.
- [11] 毛澍芬,沈世明,射影几何,上海科技文献出版社,1985.

A METHOD OF RECOVERING SHAPE FROM CONTOUR

CHENG YIMIN SUN QIBIN WANG MIN

(University of Science and Technology of China)

ABSTRACT

A method of recovering surface shape from the contour is presented. It is based on a graph and a recovery tree. Also some new algorithms of computing surface orientation from two vanishing points, one vanishing point, and without vanishing point (under certain condition), are presented. To illustrate the method, a simple example is given.

Key words: Recovery tree; perspective projection; quadrangle; surface orientation.