

基于轮廓线匹配的二维重迭物体的识别

张 鸿 宾

(北京计算机学院)

摘 要

本文提出一种基于轮廓线匹配的二维重迭物体的识别算法,并采用一种点模式匹配的快速算法进行特征点序列的匹配,通过引进形状移位数的概念和利用问题中的各种约束条件,提高了物体识别速度。

关键词: 计算机视觉,物体识别,形状匹配,点模式匹配。

一、引 言

二维物体,特别是二维重迭物体的识别问题,作为计算机视觉中一个重要的研究课题,迄今为止已有许多人作了大量的研究工作。由于在实际问题中,模型物体的形状是预先知道的,模型物体的数目也有限,因此可预先把模型物体的特征存储在模型数据库中,建立所谓基于模型的识别系统。文献[1]就是有关这方面问题的一个综述。

重迭物体的识别问题和单一物体的识别相比,由于问题的特殊性,在方法上有许多不同之处。在单一物体的识别中,人们常常使用表示物体形状的全局性特征,如区域的重心和面积,边界线的长度以及表示区域形状特征的各种度量如 compactness 等。此外,链码、傅里叶描述符和各种形式的矩也是经常使用的特征。实践表明,使用全局性特征对单一物体的识别是行之有效的方法。但是,在若干物体相互重迭或者接触的情况下,全局性特征已经不能再继续使用。其原因主要有以下两点:

- 1) 在物体相互接触或重迭的情况下,单一物体的特征在复合图象中常常消失。
- 2) 在物体相互重迭或接触的复合图象中,经常出现一些假的特征。

为了解决这些问题,有效的方法是使用物体的局部性特征。许多人使用松弛法、霍夫变换法,匹配图或树的搜索法等来解决这个问题^[2-7]。总的看来,目前已有算法的识别能力、特别是识别速度还需要进一步提高以适应实际应用的需要。

重迭物体的识别大致可以分为图象处理和形状匹配两个阶段。即使假定图象处理的方法十分完善,图象匹配的问题由于计算量大,也是相当花费时间的。实际上,在图象处理阶段,由于噪声和图象处理方法等因素的影响,经常会出现特征的误抽出和脱落等问题,这更加重了图象匹配时的负担。对于一个实用的二维重迭物体的识别系统来说,迫切

需要一种快速的图象处理和形状匹配的算法。本文提出一种基于轮廓线匹配的识别算法。这种算法可以识别扁平一类的物体,或者是有少数个固定位置的三维物体,其中每个位置都可以用一个不同的二维模型来表示。二维物体的形状首先通过轮廓线的特征点序列来表示,然后采用一种有序点模式的快速匹配算法,进行特征点序列的匹配。通过引入形状移位数的概念和利用问题中的各种约束,本文的算法在保证匹配准确性的同时,大幅度地提高了匹配的速度。和有关算法进行比较的结果说明,本算法在避免无益的搜索、缩短匹配时间上取得了相当明显的效果。

二、图象的前处理

二维物体的形状可以用轮廓线来表示。轮廓线又可以用多边形来进行有效的近似。从摄像机输入的物体的图象,首先进行二值化和边缘抽取,然后通过类似于 Pavlidis 和 Horiwits 的分一合算法^[8,9],抽取边界线并进行多边形逼近。边界线的圆弧部分可以通过圆弧抽取算法进行抽出^[10]。本文把多边形中曲率大于某个阈值的顶点和圆弧的两个端点作为轮廓线的特征点。

通过以上的处理,模型物体 U 和复合物体 V (以下也称为 U 模式和 V 模式或简称为 U 和 V), 可以分别用下面两个特征点的集合来表示

$$U = \{u_1, u_2, \dots, u_i, \dots, u_m\}, 1 \leq i \leq m,$$

$$V = \{v_1, v_2, \dots, v_j, \dots, v_n\}, 1 \leq j \leq n.$$

m 和 n 分别为两个模式的特征点的个数。为了方便以后的处理, U 和 V 的特征点都沿着轮廓线以相同的方向排列。

三、匹配误差、基准线对和形状移位数的定义

由于图象处理方法和噪声等各种因素的影响,从图象中抽出的特征点,不仅会有误抽出和漏抽出等情况,在位置上也有一定的误差。因此,把 U 模式和 V 模式进行匹配时,不是判断两个模式是否完全相同,而应当在某种误差度量的尺度下,找出误差最小的作为匹配结果。

下面先引入 U 模式和 V 模式间基准线对的定义,然后定义 U 和 V 间匹配的误差。

定义 1. 如图 1 所示,如果 U 模式的有向线段 $\overrightarrow{u_{i-1}u_{i+1}}$ 和 V 模式的有向线段 $\overrightarrow{v_{j-1}v_{j+1}}$ 满足以下的两个条件,则称它们为 U 模式和 V 模式的一个基准线对 (L_i^u, L_j^v) 。

1) 设 $|\overrightarrow{u_{i-1}u_{i+1}}| / |\overrightarrow{v_{j-1}v_{j+1}}| = F$, 那么应满足

$$(|\overrightarrow{u_i u_{i+1}}| - F \cdot |\overrightarrow{v_j v_{j+1}}|) / |\overrightarrow{u_i u_{i+1}}| < L_i,$$

$$(|\overrightarrow{u_{i-1} u_i}| - F \cdot |\overrightarrow{v_{j-1} v_j}|) / |\overrightarrow{u_{i-1} u_i}| < L_i.$$

式中 L_i 为相对长度误差的阈值。

2) 角度的差 $|\angle u_{i-1}u_i u_{i+1} - \angle v_{j-1}v_j v_{j+1}| < A_i$, A_i 为角度误差阈值。

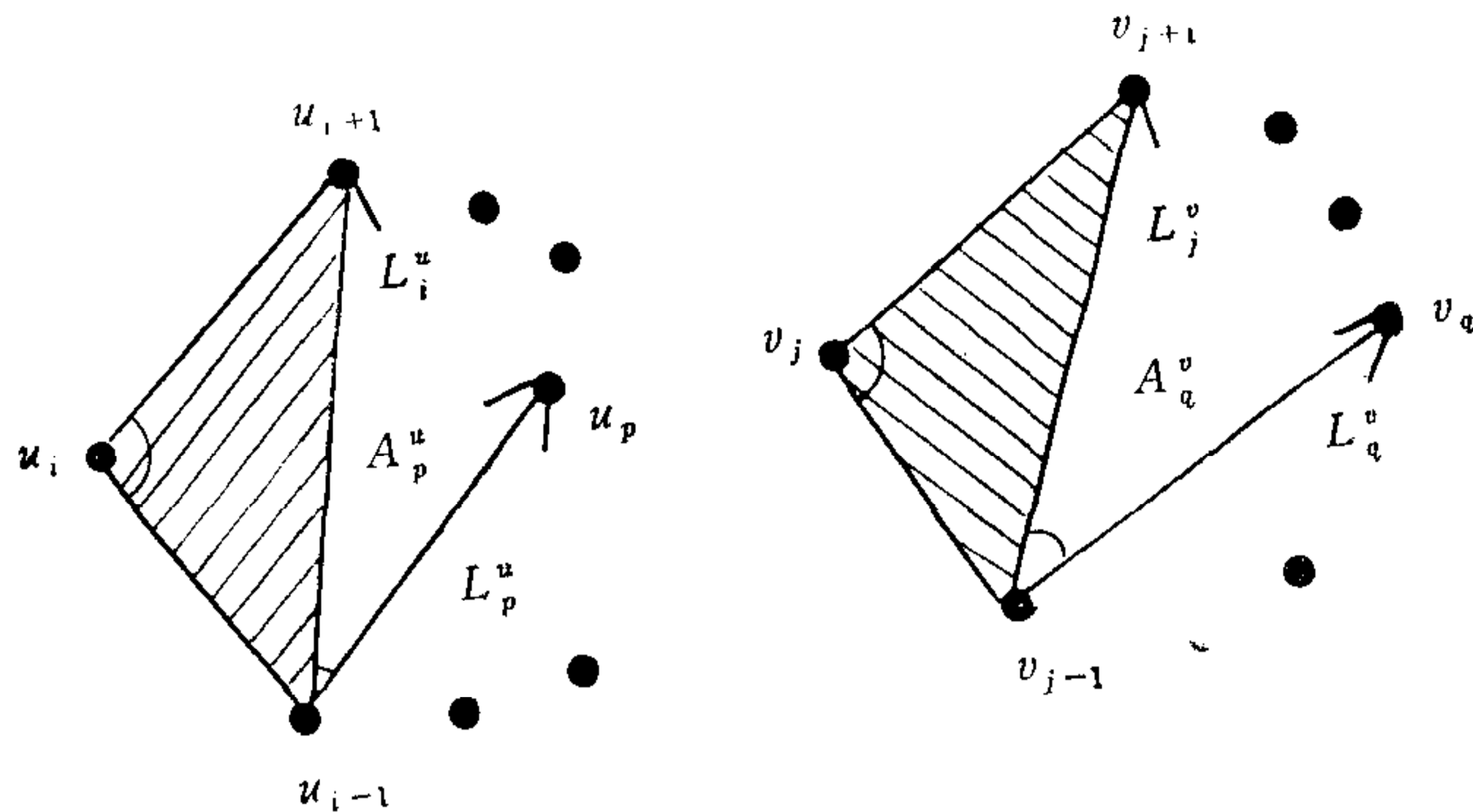


图1 基准线对的定义

条件 1) 中的 F 称为 U 和 V 间的比例因子。三角形 $\Delta u_{i-1}u_iu_{i+1}$ 和 $\Delta v_{j-1}v_jv_{j+1}$ 称为 U 模式和 V 模式的基准三角形。

在满足上面的条件 1), 且当 $|\angle u_{i-1}u_iu_{i+1} - \pi|$ 较大时, 上面的条件 2) 和下面的 2') 等价。

2') u_i 和 v_j 在有向线段 $\overrightarrow{u_{i-1}u_{i+1}}$ 和 $\overrightarrow{v_{j-1}v_{j+1}}$ 的同一侧。

由于 $|\angle u_{i-1}u_iu_{i+1} - \pi|$ 的值较小的特征点已在图象的前处理阶段被合并, 所以可以用条件 2') 来代替 2)。条件 2') 的检查在计算上要方便。

定义 2. 假定 (L_i^u, L_j^v) 为 U 和 V 的一个基准线对。当把 U 模式的点 u_p 和 V 模式的点 v_q 相匹配时, 由于这一对点的匹配而产生的匹配误差按如下式计算(参考图 1)。

设 u_p 和 u_{i-1} 间的距离为 L_p^u , $u_{i-1}u_p$ 和 $u_{i-1}u_{i+1}$ 间的角度为 A_p^u 。同样, 设 v_q 和 v_{j-1} 间的距离为 L_q^v , $v_{j-1}v_q$ 和 $v_{j-1}v_{j+1}$ 间的角度为 A_q^v 。这时, 相对于基准线对 (L_i^u, L_j^v) , 把 u_p 和 v_q 相匹配时所产生的误差定义为

$$w(u_p, v_q; L_i^u, L_j^v) = KA \cdot |A_p^u - A_q^v| + KL \cdot |L_p^u - F \cdot L_q^v|.$$

式中的 KA 和 KL 分别为角度误差和长度误差的加权因子。

前面定义比例系数 F 的目的是为了适应两个模式间的比例变化。使用点间的相互距离和相互角度, 保证了算法和物体的平移及旋转无关。

通过上面的定义可以看出, 基准线对的定义不仅为匹配误差的度量提供了一个基础, 而且由于基准线对使用了轮廓形状的局部约束条件, 它也为以后整个形状的匹配提供了一个出发点。

根据定义, U 和 V 间的基准线对最多可以有 $m \cdot n$ 个, 然而其中含有大量的冗长性。对于 U 模式和 V 模式的一种匹配状态来说, 只要找出一个基准线对就可以了。为了表示 U 和 V 间不同的对应状态, 下面我们引入形状移位数的概念。

定义 3. 基准线对 (L_i^u, L_j^v) 的形状移位数 SSN 定义为

$$SSN = i - j,$$

$$\text{IF}(SSN.LT.0) \quad SSN = SSN + n.$$

可以证明, U 和 V 的同一个对应状态的不同基准线对的形状移位数是相同的, 而形状

移位数相等的不同基准线对也出自同一个 U 和 V 的对应状态。因此,在以后的匹配中,对形状移位数相等的基准线对,只要检查一个就可以了。这样,必须检查的基准线对最多有 n 个。此外,由于基准线对还必须满足反映局部约束的两个条件,因此对于实际问题来说,算法中所实际检查的基准线对数 K 是远小于 n 的,这一点也为后面的实验所证实。

四、轮廓线特征点的匹配

有关点集合间匹配的研究结果表明,当有特征点的插入和脱落时,匹配算法要复杂得多,计算量也大得多。为了适应实用化的要求,缩短轮廓线特征点间匹配的时间,本文采用了一种有序点模式的快速匹配算法。下面简单描述该算法的原理^[1]。

设 (L_i^u, L_i^v) 为 U 和 V 的一个基准线对。对基准三角形以外的各点,根据它们和基准线的角度大小,以升序的次序构成两个有序表 UL 和 VL 。

$$UL = \{u_1, u_2, \dots, u_p, u_{p+1}, \dots, u_m\},$$

$$VL = \{v_1, v_2, \dots, v_q, v_{q+1}, \dots, v_n\}.$$

假定现在把 u_p 和 v_q 作了匹配,那么下一个匹配点对的选择,自然是 u_{p+1} 和 v_{q+1} 或 v_{q+1} 以后的一定范围以内的某个点。设 u_{p+1} 和基准线的角度为 A_u , 容许角度误差为 A_t 。那么,只有满足 $|A_u - A_v| < A_t$ 的 VL 中的点(它和基准线的角度为 A_v),才有可能和 u_{p+1} 进行匹配。这样就限制了匹配时搜索的范围,避免了盲目的点的组合。对于实际问题中的最大允许误差 A_t 来说,和 UL 各元素间有可能匹配的 VL 的元素数就限制在某一常数范围内。因此,在特征点的误差范围互不重迭的情况下, UL 和 VL 间的匹配可以在 UL 元素数的线性时间内完成。

特征点 u_p 和 v_q 必须满足下面的三个条件才能进行匹配(参考图 1)。

条件 1. 设 u_p 和基准线 L_i^u 间的角度为 A_p^u , v_q 和基准线 L_i^v 的角度为 A_q^v 。那么, $A_e = |A_p^u - A_q^v|$ 应小于最大允许角度误差 A_t 。

条件 2. 设 u_p 和基准线起点间的距离为 L_p^u , v_q 和基准线起点间的距离为 L_q^v 。那么,相对长度误差 $L_e = |L_p^u - F \cdot L_q^v| / L_p^u$ 应小于某一阈值 L_t 。

条件 3. u_p 和 v_q 同它们相邻两点间的关系,必须满足和定义基准线对时同样的约束条件。

在匹配时,由于特征点的插入和脱落,可能会出现 U 的某个特征点和 V 的任何点都不能匹配的情况(同样, V 也有这种情况)。这时可引入符号 (u_p, Λ) 或 (Λ, v_q) 。此时的匹配误差按 $KA \cdot A_t + KL \cdot L_e$ 计算。式中 A_t 为最大允许角度误差, L_e 为 u_p (或 v_q) 和基准线起点间的距离(对于 v_q 来说,是乘上比例因子 F 后的长度)。没有对应点也计算误差的目的,是为了防止有对应点时的误差反而比没有对应点时的误差更大的现象发生。

五、实验结果

在日本 NEC 制造的超小型计算机 MS-190 上,使用各种模拟和实际的图象数据

对上述算法进行了实验。下面是其中的一部分实验及其结果。

实验 1. 为了检验上述算法是否正确和有效,使用随机发生的图象数据作了试验。首先用二维均匀随机数发生程序产生 n 个随机点,作为 V 的特征点。然后从中任取 $m(m \leq n)$ 个点,在旋转一定的角度、坐标放大或缩小以及加进一定的随机噪声后,作为 U 的特征点。即

$$\begin{aligned}x' &= (x \cdot \cos \beta - y \cdot \sin \beta) \cdot (f + c), \\y' &= (x \cdot \sin \beta + y \cdot \cos \beta) \cdot (f + c).\end{aligned}$$

式中 (x, y) 是 V 的特征点的坐标, (x', y') 是对应的 U 的特征点的坐标。 β 是旋转角度, f 是放大缩小因子, c 是均值为 0、方差为 σ^2 的正态随机数。

通过给二维随机数发生程序不同的初始值,产生了 8 种、每种各 10 组的不同特征点序列。实验中的参数为 $L_i = 0.15$, $A_i = 10^\circ$, $\sigma = 0.025$ 。对于所有这些数据,算法都给出了正确的匹配结果。匹配时 CPU 时间的平均值和基准线对数的平均值见表 1。从表中可以看出,对几十点规模的问题(这对一般的形状匹配问题是足够了),算法的结果是相当令人满意的。从表中也可以看出,基准线对的个数远少于特征点的个数。

实验 2. 图 2 的 (c) 是 (a) 和 (b) 重迭后的模式。在 (c) 中, (a) 模式的部分被扩大了某一倍数。(a)、(b) 和 (c) 间匹配的结果如图 3 所示。匹配上的点用圆圈表

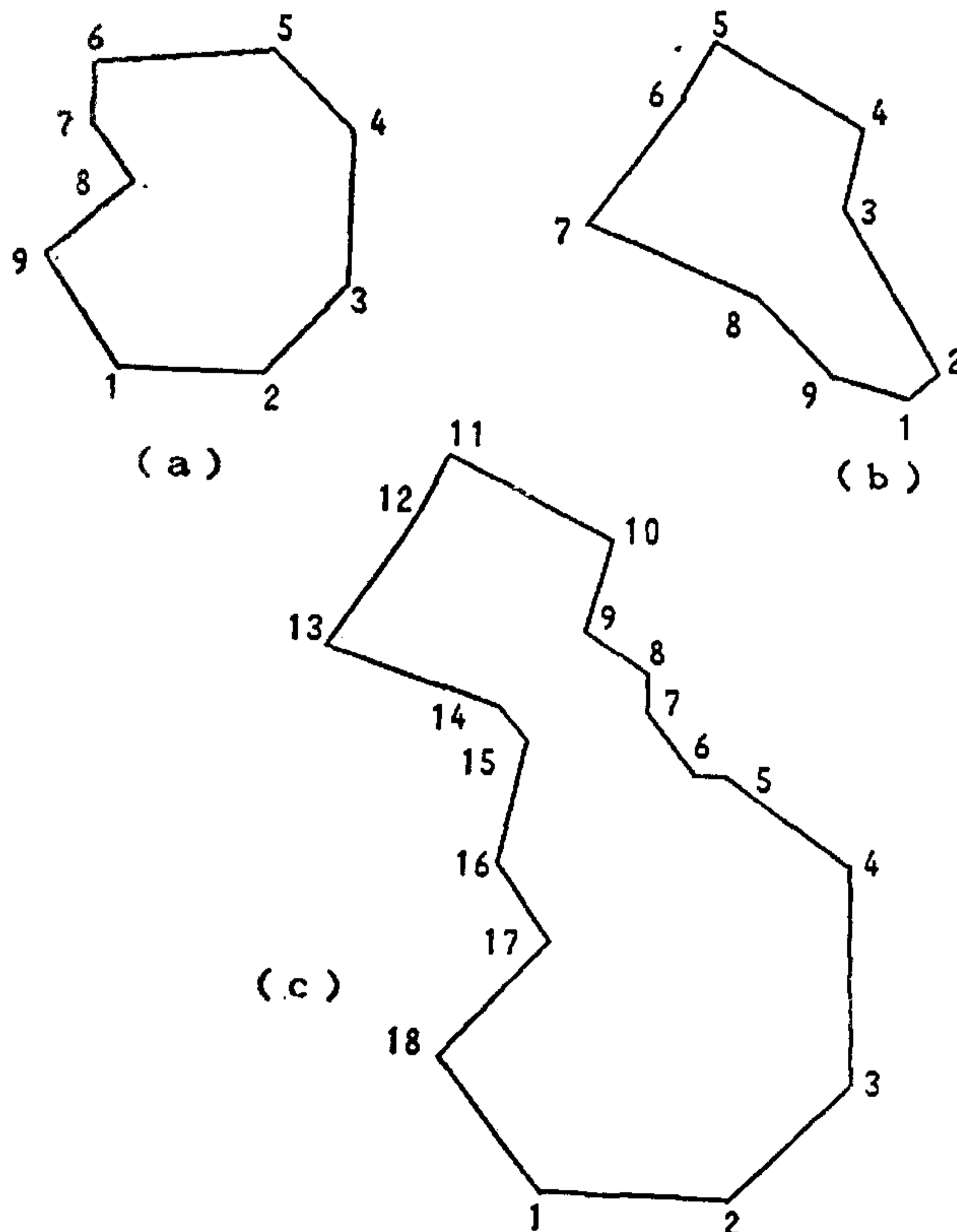


图 2 二维重迭物体的识别(实验 2)

表 1 平均 CPU 时间 (ms) 和平均基准线对数

U 的点数	30	30	50	50	70	70	90	100
V 的点数	30	40	50	60	70	80	90	100
平均 CPU 时间	86	113	318	380	810	835	2335	4042
平均基准线对数	2.1	2.1	5.7	3.8	7.0	5.3	12.7	19.4

示,没有匹配上的用小三角形表示(以下同)。这个实验的数据是从参考文献[3]中引用来的。虽然实验的环境不同,不便进行准确的比较,但本文算法的结果看来要好。表 2 是本实验和文献[3]的实验环境及所用 CPU 时间的比较表。

表 2 本算法和文献[3]的比较

	匹配的 CPU 时间		使用的计算机
	a 和 c	b 和 c	
计算法	25ms	23ms	MS-190, 4.0MPS
文献 [3]	52.79s	42.90s	PDP-10, 1.8MPS

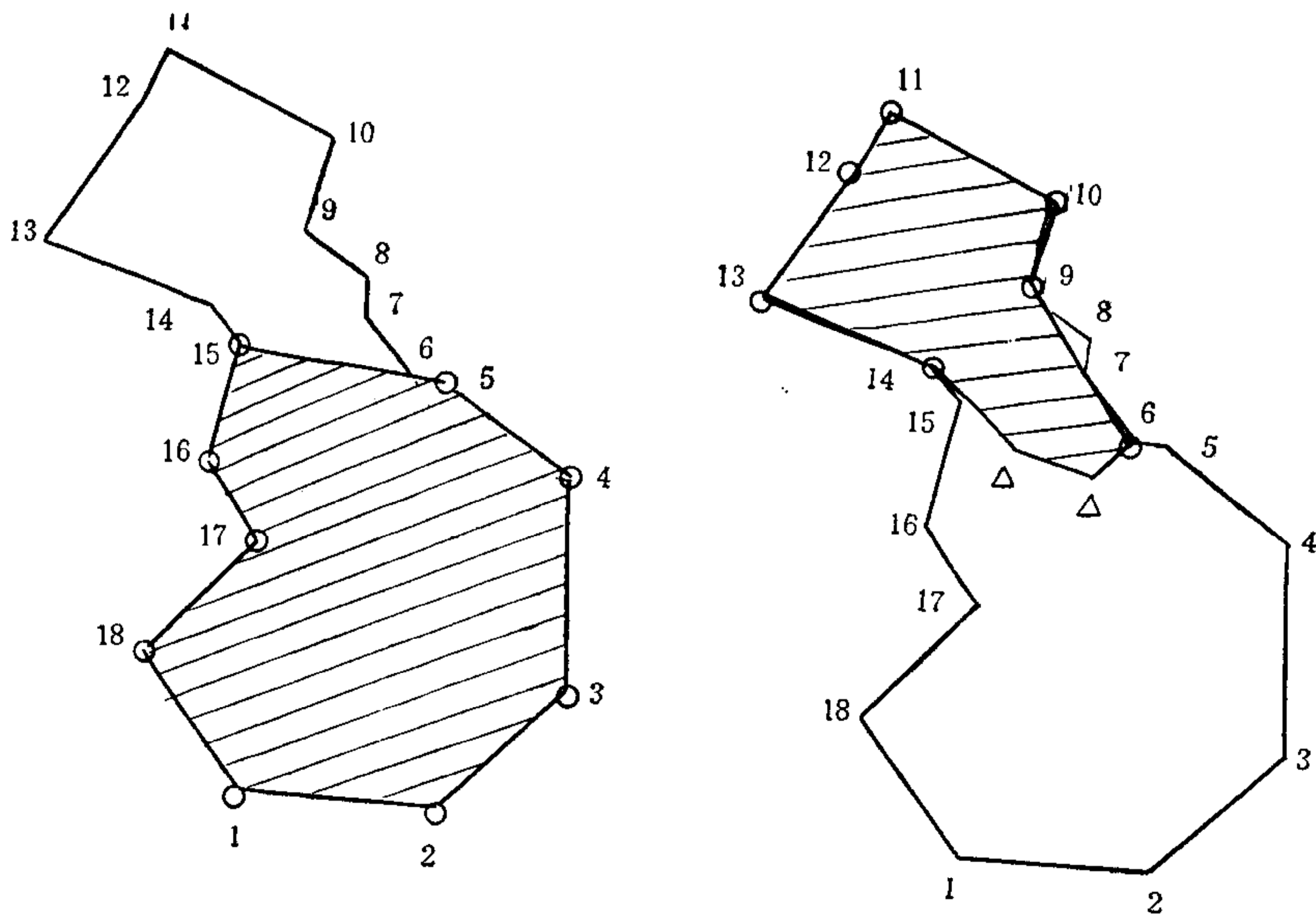


图 3 实验 2 的结果

实验 3. 图 4 是一种 Jig-saw Puzzle 问题。把图 4(a)、(b)、(c) 的物体重迭,形成了图 4(d) 那样的模式。本算法从复合模式图 4(d) 中正确地识别出了部件的图 4(a)、(b) 和 (c)。匹配的结果如图 5 所示。总匹配时间是 66ms。这个实验的数据也是从参考文献[3]中引用来的。文献[3]所用的匹配时间为 152.7s。

实验 4. 图 6 的 (a) 和 (b) 是工件的图象经过边缘抽取和多边形近似后的图形,图 6(c) 是 (a) 和 (b) 重迭后的复合物体的图象经过边缘抽取和多边形近似后的图形。图

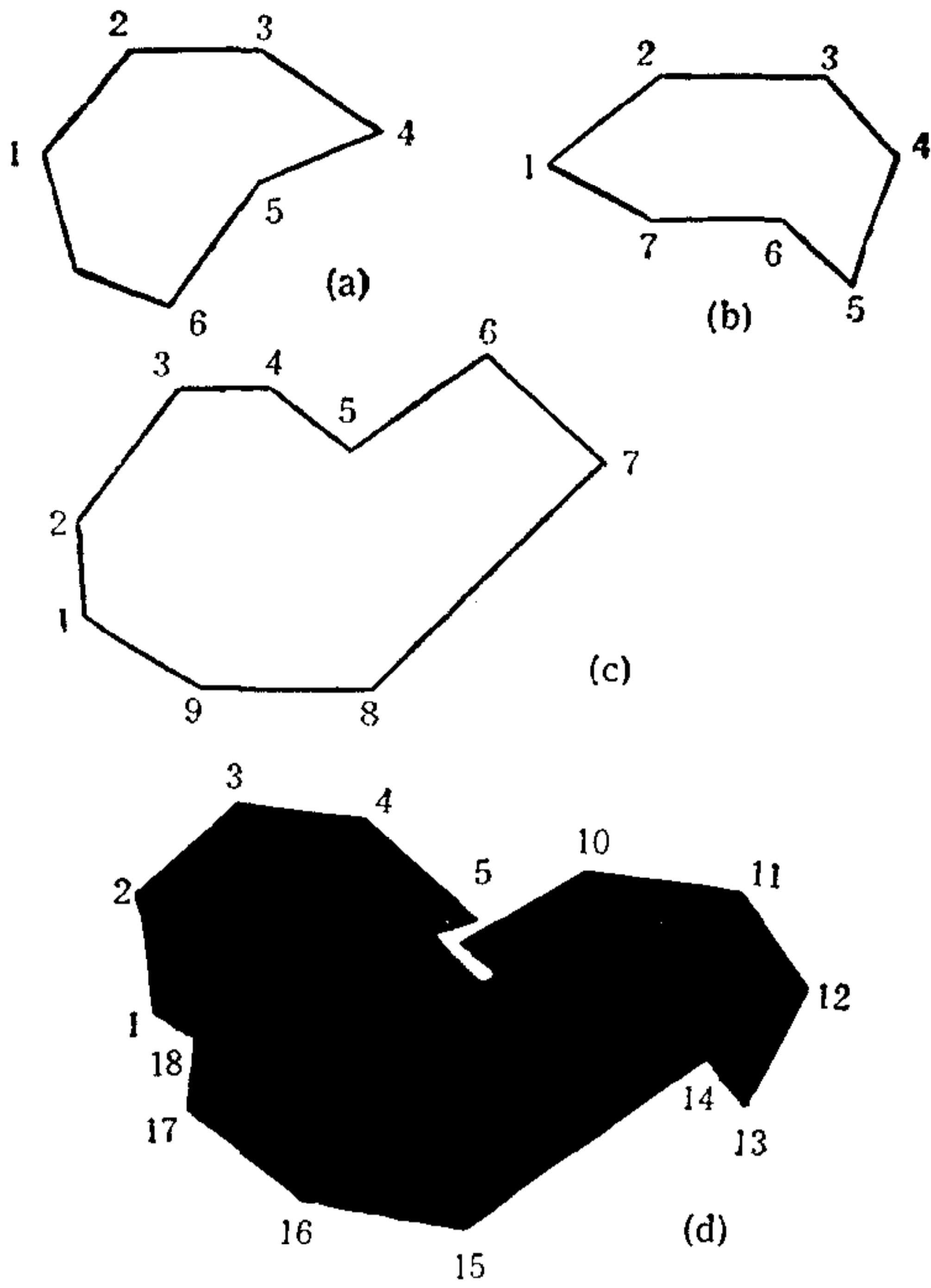


图4 Jig-saw Puzzle 问题(实验3)

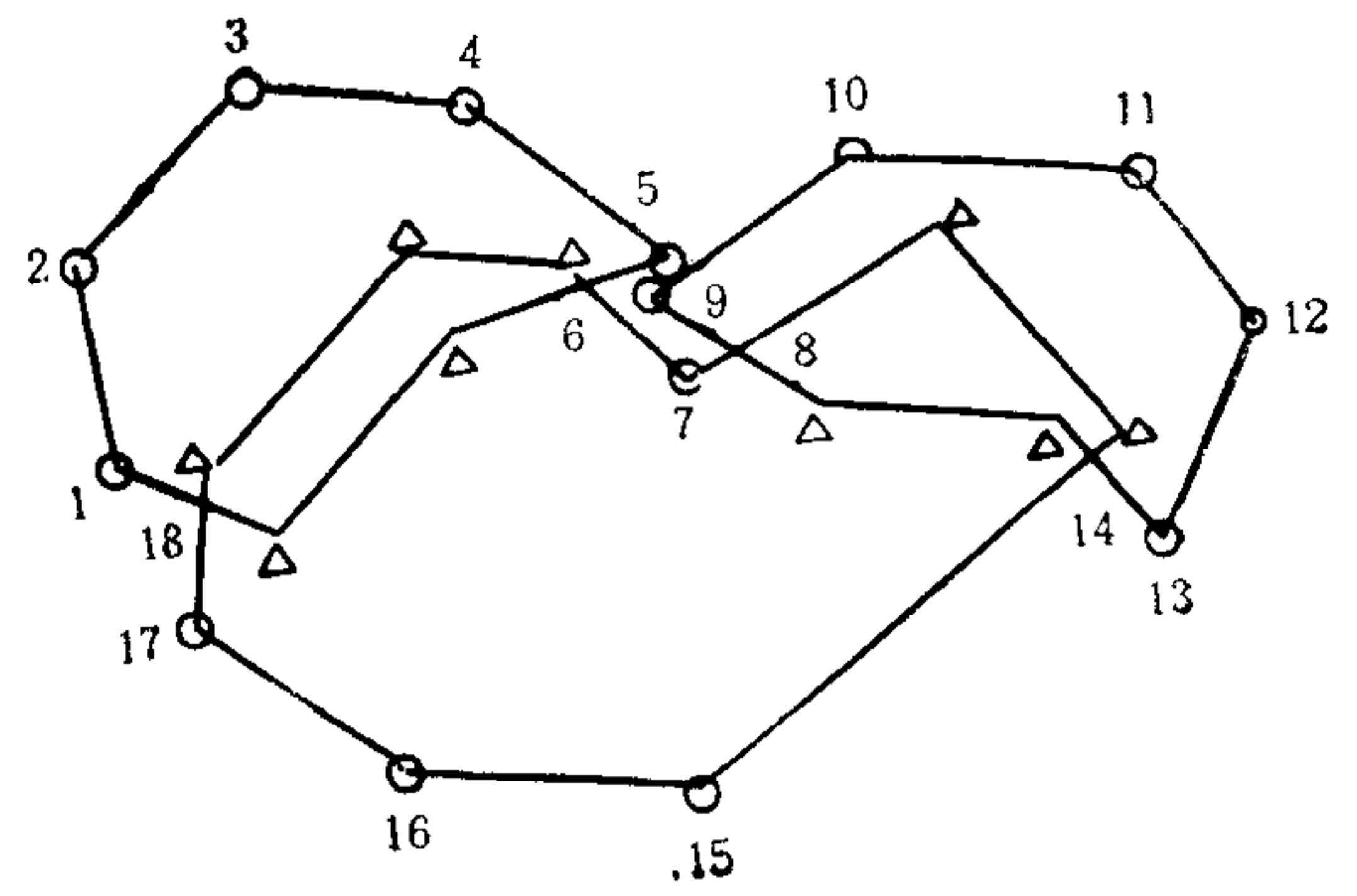


图5 实验3的匹配结果

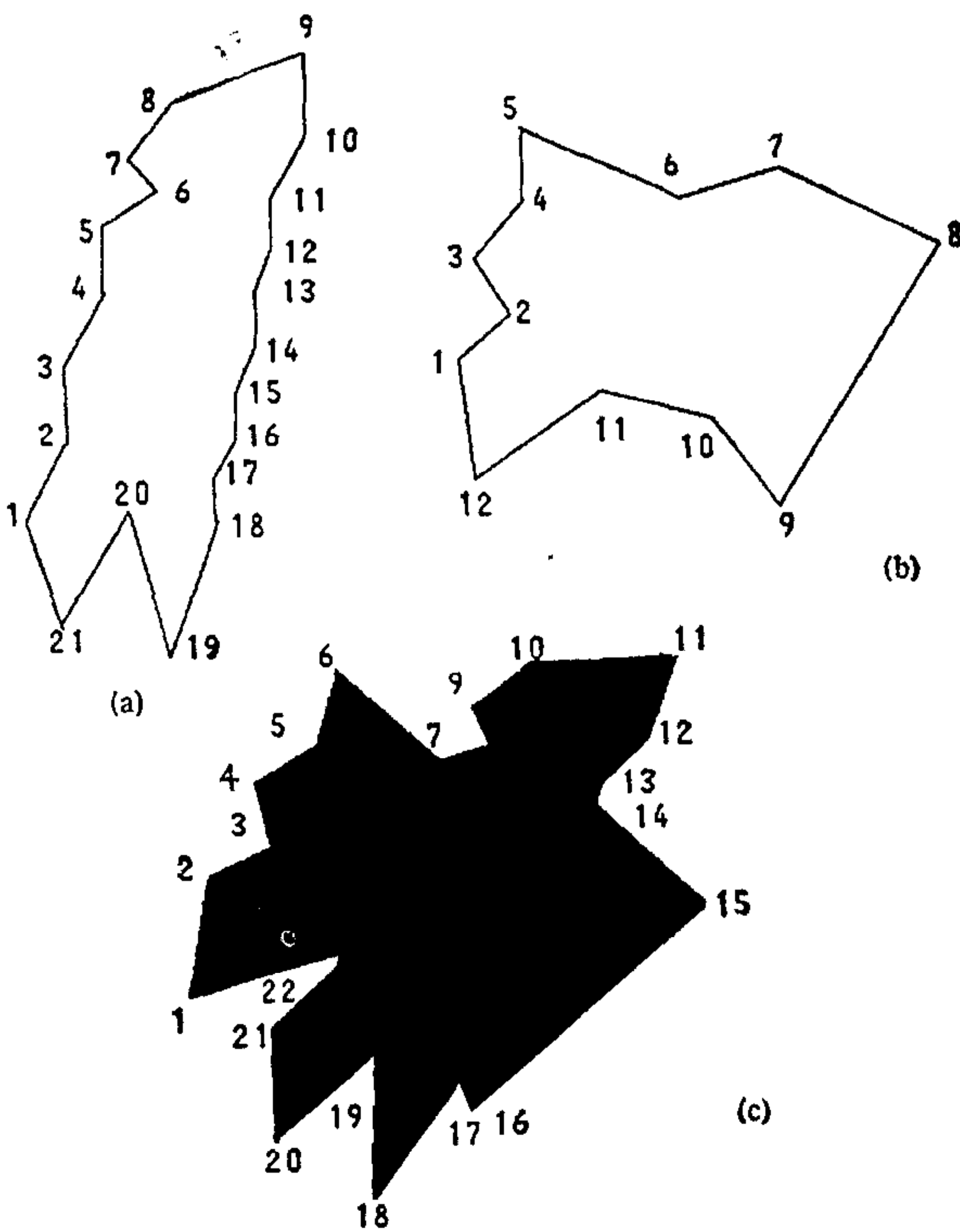


图6 工件图象多边形近似后的图形(实验4)

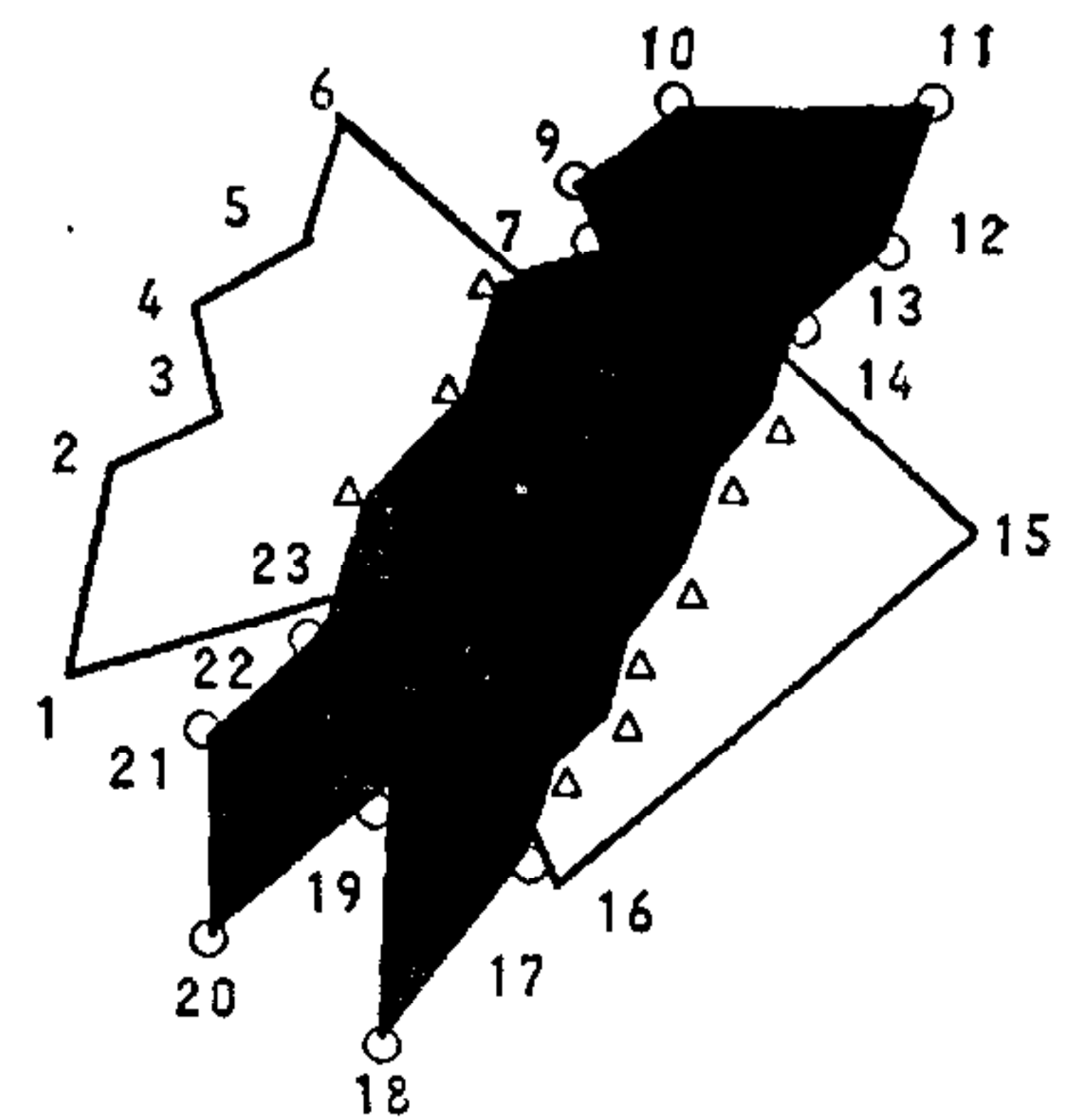


图7 (a) 和 (c) 匹配的结果(实验4)

6(a) 和 (c) 匹配的结果如图 7 所示。其中图 7(a) 的特征点数为 21 点, 图 7(c) 的特征点数为 23 点。匹配时间是 63ms。

实验结果以及和某些算法相比较的数据表明, 本文算法在保证匹配准确性的同时, 大大提高了匹配的速度, 显示了实际应用的可能性。

参 考 文 献

- [1] Chin, R. T. and Dyer, C. R., Model-based Recognition in Robot Vision, *ACM Comput. Surveys*, 18-1 (1986), 67—108.
- [2] Davis, L. S., Shape Matching Using Relaxation Techniques, *IEEE PAMI*, **PAMI-1**(1979), 1, 60—72.
- [3] Bhanu, B. and Faugeras, O. D., Shape Matching of Two-dimensional Objects, *IEEE PAMI*, **PAMI-6**(1984), 2, 137—156.
- [4] Ayache, N. and Faugeras, O. D., HYPER: A New Approach for the Recognition and Positioning of two-dimensional Objects, *IEEE PAMI*, **PAMI-8**(1986), 1, 44—54.
- [5] Koch, M. W. and Kashyap, R. L., Using Polygons to Recognize and Locate Partially Occluded Objects, *IEEE PAMI*, **PAMI-9**(1987), 4, 483—494.
- [6] Turney, J. L., Mudge, T. N., and Volz, R. A., Recognizing Partially Occluded Parts, *IEEE PAMI*, **PAMI-7**(1985), 4, 410—421.
- [7] Grimson, W. ERIC L. and Lozana-perez, T., Localizing Overlapping Parts by Searching the Interpretation Tree, *IEEE, PAMI*, **PAMI-9**(1987), 4, 469—482.
- [8] Pavlidis, T., Piecewise Approximation of Plane Curve, Proc. 1st IJ CPR, 1973, 396—405.
- [9] Pavlidis, T., and Horowitz, S., Segmentation of Plane Curves, *IEEE Comput.*, **C-23**(1974), 860—870.
- [10] Pavlidis, T., Algorithms for Graphics and Image Processing, Computer Science Press, 1982.
- [11] Zhang, H., Minoh, M., Ikeda, K., Fast Algorithm for Point Pattern Matching by Using Ordered list, Trans. of IPSJ (Japan), 1990, 7, 1005—1014.

THE RECOGNITION OF 2-D OVERLAPPING OBJECTS BY BOUNDARY MATCHING

ZHANG HONGBIN

(AI Research Lab., Beijing Computer Institute, Beijing, 100044)

ABSTRACT

A method for recognizing 2-D overlapping objects by boundary matching is presented. A fast algorithm of point pattern matching is adopted for the matching of feature point sequence. And a concept of shape shift number is introduced to reduce the amount of search while matching so that the recognition speed can be increased and matching time be shortened.

Key words: Computer vision; object recognition; shape matching; point pattern matching.