

文章编号 1004-924X(2010)06-1413-07

应用 Mean Shift 和分块的抗遮挡跟踪

颜 佳, 吴敏渊, 陈淑珍, 张青林

(武汉大学 电子信息学院, 湖北 武汉 430079)

摘要: 针对传统 Mean Shift 跟踪算法在目标发生遮挡时容易跟偏甚至跟丢的缺陷, 提出了一种新的抗遮挡跟踪算法。首先, 对跟踪窗口内的目标进行分块; 然后, 对外围子块分别实施 Mean Shift 跟踪算法并检测遮挡的发生, 当遮挡发生后即对所有子块实施 Mean Shift 跟踪算法; 最后, 引入一种子块置信度机制并仅用置信度最高的子块来确定目标的最终位置, 从而在目标发生遮挡时能有效剔除被遮挡子块对目标定位的影响。对不同的视频序列测试的结果显示, 本算法能对发生遮挡的目标进行准确跟踪。当遮挡目标尺寸为 70 pixel \times 100 pixel 时, 平均处理时间为 38.6 ms/frame。结果表明, 改进算法能够满足目标跟踪系统稳定性和实时性的要求。

关键词: Mean Shift; 目标跟踪; 分块; 抗遮挡

中图分类号: TP391 **文献标识码:** A **doi:** 10.3788/OPE.20101806.1413

Anti-occlusion tracking algorithm based on Mean Shift and fragments

YAN Jia, WU Min-yuan, CHEN Shu-zhen, ZHANG Qing-lin

(School of Electronic Information, Wuhan University, Wuhan 430079, China)

Abstract: A new anti-occlusion tracking algorithm is presented to solve the problem that traditional Mean Shift based tracking algorithm often deviates or loses the targets under occlusions. Firstly, the target in the tracking window is divided into a number of fragments, then the Mean Shift algorithm is used to move the peripheral fragments separately and to detect the occlusions. Furthermore, all fragments can be computed when the target is occluded. Finally, the confidence of each fragment is computed and only the fragment with the highest confidence is involved to achieve the whole target's coordinates and to avoid the influence of occluded fragments on the target location. Tested results for different video sequences indicate that proposed algorithm can track precisely the target occluded. When the occlusion size is 70 pixel \times 100 pixel, the running time is 38.6 ms/frame, which meets the requirements of target tracking system for the stability and real time.

Key words: Mean Shift; target tracking; fragment; anti-occlusion

1 引言

运动目标实时跟踪在计算机视觉领域有着非常重要的应用。Mean Shift 算法是一种基于密度梯度的无参数估计方法^[1],因其迭代次数少计算量小已经广泛应用在目标实时跟踪中^[2-4]。但使用该方法对发生遮挡的目标进行跟踪时,由于其没有有效的抗遮挡机制,往往会导致目标的跟偏甚至跟丢。

针对此问题,文献[5]研究了粒子滤波器和 Mean Shift 算法相结合的情况,首先使用 Mean Shift 算法跟踪目标,然后使用粒子滤波器来降低遮挡的干扰以确定真正的目标中心坐标,但粒子滤波器自身的复杂计算降低了跟踪的实时性。文献[2]和[6]中都提出了在发生遮挡时用 Kalman 滤波的方法来预测目标位置,但该方法在长时间遮挡时失效。文献[7]通过将运动目标分块化来处理遮挡情况,但该方法在复杂环境下稳定性不够,而且对各子块未能赋予有效的权重。在解决遮挡问题时,将目标分块化是一种常用的方法。例如:文献[8]使用了大量的分块,并且采用穷搜索的办法来跟踪目标,尽管采取了积分直方图的办法来降低计算复杂度,但当目标尺寸稍大时仍很难保证实时性;文献[9]在得到人体的轮廓模型的前提下将人体分为头部、上半身和下部分三子块来实施跟踪,但有时并非容易得到目标的轮廓模型。

鉴于分块在处理遮挡问题时的优越性和 Mean Shift 跟踪算法的实时性,本文将分块思想引入到文献[4]中的 Mean Shift 框架中来,也可以说是将 Mean Shift 算法引入到文献[8]的分块模型之中。该算法首先对跟踪窗口内的目标进行分块,然后对外围子块实施 Mean Shift 跟踪算法,在独立寻找局部最优解的同时检测遮挡的发生,当遮挡发生后将所有子块都纳入计算并利用置信度最高的子块来确定整体目标的最终坐标位置,所以当目标发生遮挡时该算法能有效剔除被遮挡子块对目标定位的影响。实验表明,本文提出的抗遮挡跟踪算法能很好地定位跟踪目标并且实时运行。

2 Mean Shift 跟踪算法的缺陷

Mean Shift 算法是一种半自动化跟踪算法。首先通过手工选定目标,计算核函数加权下的跟踪窗口中的直方图分布。假设 $\{y_i\}_{i=1,\dots,n}$ 为目标区域中各个像素的位置,目标中心为 x_0 ,则目标的直方图分布为:

$$\hat{p} = \{\hat{p}_u\}_{u=1,\dots,m}, \quad (1)$$

其中: $\hat{p}_u(x_0) = C \sum_{i=1}^n G\left(\left\|\frac{y_i - x_0}{h}\right\|^2\right) \delta[b(y_i) - u]$, G 为核函数, m 为特征空间中特征值的个数(本文 G 取高斯核函数,特征空间采用 HSV 色彩空间中的 H 、 S 信息, m 为 $16 \times 16 = 256$ 级), δ 为 Kronecker 函数, $b(y_i)$ 为像素 y_i 对应的特征值, C 为归一化系数, h 为核函数的带宽,一般设为跟踪窗口的一半。

其后的跟踪过程就是在序列图像中通过迭代寻找与目标最为相似的候选区域,相似性度量采用 Bhattacharyya 系数。目标模型的概率分布 $\hat{p}(x_0)$ 与候选区域概率分布 $\hat{q}(x)$ 的 Bhattacharyya 系数为:

$$\rho(x) = \sum_{u=1}^m \sqrt{\hat{p}_u(x_0) \hat{q}_u(x)}. \quad (2)$$

对应的迭代公式为:

$$x_1 = \frac{\sum_{i=1}^n y_i \omega(y_i) G\left(\left\|\frac{y_i - x_0}{h}\right\|^2\right)}{\sum_{i=1}^n \omega(y_i) G\left(\left\|\frac{y_i - x_0}{h}\right\|^2\right)}, \quad (3)$$

x_1 为新的目标中心位置,其中:

$$\omega(y_i) = \sum_{u=1}^m \sqrt{\frac{\hat{q}_u(x)}{\hat{p}_u(x_0)}} \delta[b(y_i) - u], \quad (4)$$

$\omega(y_i)$ 为权重函数。迭代过程就是不断计算式(3),直至 Bhattacharyya 系数最大,即定位为最终中心位置,停止迭代。

图 1 给出了传统 Mean Shift 算法(tracker MS)对文献[8]中一段视频序列 Face 进行跟踪的部分结果(初始跟踪窗口为手工选定,其大小为 $70 \text{ pixel} \times 100 \text{ pixel}$)。由于传统 Mean Shift 算法中未提供解决遮挡问题的办法,所以当目标左脸部进入遮挡时(第 47 帧),由于遮挡物的影响会导致目标的跟偏,跟踪窗口定位于右脸之上;而当大部分目标被遮挡时(第 357 帧),因该视频序列中颈部和脸部颜色特征类似所以跟踪窗口会定位于颈部之上导致目标跟丢。



图 1 Tracker MS 对 Face 视频序列的跟踪结果(帧 1、18、47、134、211、279、322 和 357)

Fig. 1 Tracking results of the Face video sequence using tracker MS(Frame 1,18,47,134,211,279,322 and 357)

3 基于分块的跟踪方法

为了增强 Mean Shift 算法的抗遮挡能力,本文引入了分块的思想。

算法初始化时需将目标分成若干子块。由于子块数目过多会影响计算的实时性^[8],子块数目

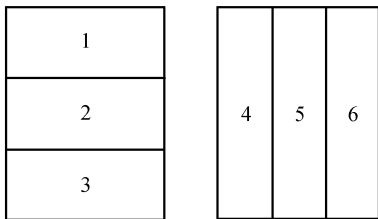


图 2 分块示意图

Fig. 2 Scheme of fragments

过少会降低抗遮挡能力,而子块数目降为一块时即变为原始的 Mean Shift 算法。所以本文采取如图 2 所示的分块方法,其中子块的数目 T 取为横向 3 块和纵向 3 块,共 6 块。各子块中心相对于目标中心位置偏移量定为 Δx^t ,其中 $t=1, \dots, T$ 。然后在目标跟踪过程中,对各子块分别实施第二部分所述的 Mean Shift 算法,这样即可获得各子块的最终位置,此时公式(3)应为:

$$x_1^t = \frac{\sum_{i=1}^{n_i} y_i \omega(y_i) G\left(\left\| \frac{y_i - x_0^t}{h_t} \right\|^2\right)}{\sum_{i=1}^{n_i} \omega(y_i) G\left(\left\| \frac{y_i - x_0^t}{h_t} \right\|^2\right)}, \quad (5)$$

其中: n_i 为各子块内的像素个数, h_t 为各子块的核函数带宽。此时各子块的 Bhattacharyya 系数

对应为:

$$\rho(x^t) = \sum_{u=1}^m \sqrt{\hat{p}_u(x_0^t) \hat{q}_u(x^t)}. \quad (6)$$

为减少计算量,最初仅计算外围子块 1、3、4 和 6。当

$$\min\{\rho(x_1^1), \rho(x_1^3), \rho(x_1^4), \rho(x_1^6)\} < th, \quad (7)$$

则表明遮挡发生,即将所有子块纳入计算。 th 为检测遮挡的阈值,该值取得越大,对遮挡越敏感,若 th 为 1,则将导致上面的手段失去意义;该值取得过小,会导致仅当遮挡严重时才将所有子块纳入计算。本文中 th 取为 0.7。

本文利用置信度最高的子块来决定整体目标的位置,而其中各子块的置信度不仅由它的相似性 Bhattacharyya 系数决定,而且包括它和背景的分度 L_t 。假设各子块的置信度为 C_t ,则

$$C_t = \rho_t + L_t, \quad (8)$$

其中: L_t 是每个子块与背景的分度, L_t 越大,说明该块与背景的分度越高。本文采用文献[10]中的分度方法:

$$L_t = \frac{\sum_{i=1}^{n_i} \log\left(\frac{\max\{\hat{p}_u(b(y_i)), \epsilon\}}{\max\{\hat{p}_b(b(y_i)), \epsilon\}}\right)}{n_i}, \quad (9)$$

$\hat{p}_b(b(y_i))$ 为子块内像素对应的颜色特征值在背景归一化直方图中的取值,本文中背景取为跟踪窗口外 15 pixel 的范围,见图 3 中虚线框所示。 ϵ 为一常数,用于避免被除数为零或对零取对数的发生。但若该值过大,则失去了区分度的意义。本文 ϵ 取为 0.1。

在各子块完成 Mean Shift 迭代后,由各自的置信度 C_t 可得其中置信度最大的子块 M 为:

$$M = \arg \max_i C_i, \quad (10)$$

然后以子块 M 为标准获得整体目标的新坐标值为 x_1 :

$$x_1 = x_1^M - \Delta x^M, \quad (11)$$

在得到目标的真实坐标后更新其它子块的坐标值以便进行下一帧的跟踪。

$$x_0' = x_1 + \Delta x' (t=1, \dots, T, t \neq M). \quad (12)$$

4 算法流程

Mean Shift 跟踪算法在目标进入遮挡时易出现跟偏和跟丢的问题,而将目标分块可以很好地检测遮挡的发生。因此本文将两者有机地结合在一个跟踪框架之中,具体算法流程如下:

(i) 初始化选定目标(本文中目标为手工选定),按图 2 中示意图将目标分成 T 个子块,并按式(1)计算各子块直方图,子块集为 $\{1, 3, 4, 6\}$;

(ii) 读取新一帧图像,对子块集中各子块利用 Mean Shift 算法求取其新坐标值:

(a) 按式(1)计算候选位置直方图;

(b) 根据式(4)求权重函数,式(5)求 x_1' ;

(c) 若 $\|x_0' - x_1'\| < \epsilon$, 则停止迭代,否则 $x_0' = x_1'$, 转到(a)继续迭代;

(iii) 按式(8)分别得到各子块的置信度 C_i ;

(iv) 通过式(10)得到置信度最大的子块编号 M ;

(v) 计算式(11)得到整体目标在此帧时的坐标位置;

(vi) 按式(12)更新各子块位置。若式(7)成立,则子块集合为 $\{1, 2, 3, 4, 5, 6\}$, 否则子块集合为 $\{1, 3, 4, 6\}$ 。跳转到(ii)处进行下一帧的处理。

5 实验结果及分析

采用本文算法(tracker MSF)对 Face 视频序列进行跟踪的部分结果如图 3 所示,其中第一帧处显示了分块情况,其余帧处仅显示了置信度最高的子块和整体目标的矩形框,如第 134 帧处子块 1 的置信度最高。可以看出,虽然视频中脸部目标不断地被遮挡,但是本文算法具备很好的抗遮挡性能。图 4 给出了两种算法分别对文献[8]中的另一段视频序列 Woman 的跟踪结果。特别是尽管第 138 帧处目标并未出现遮挡,但是由于目标下半身形变较大,所以 tracker MS 出现跟偏的情况,而 tracker MSF 在采取了分块后有效地避免了下半身对整体目标定位的影响。图 5 给出了以上两种算法以及文献[8]中算法(tracker FRG)对 Face 序列跟踪的误差曲线图。



图 3 tracker MSF 对 Face 视频序列的跟踪结果(帧 1、18、47、134、211、279、322 和 357)

Fig. 3 Tracking results of the Face video sequence using tracker MSF(Frame 1, 18, 47, 134, 211, 279, 322 and 357)

由于分块实际上是增加了参与 Mean Shift 计算的目标数,所以 tracker MSF 肯定比 tracker MS 处理时间多,但是由于子块中参与计算的像

素点要少于整体目标并且本文采取的分块策略为仅在目标出现遮挡时才让所有子块参与计算,所以也减少了一定的计算量,较单纯的基于分块的

算法 tracker FRG^[8]要小得多。图 6 给出了 3 种算法对 Face 序列的处理时间曲线图(算法均在

Intel Pentium D 的 CPU,1 G 内存配置的电脑上用 VC 6.0 编程实现)。

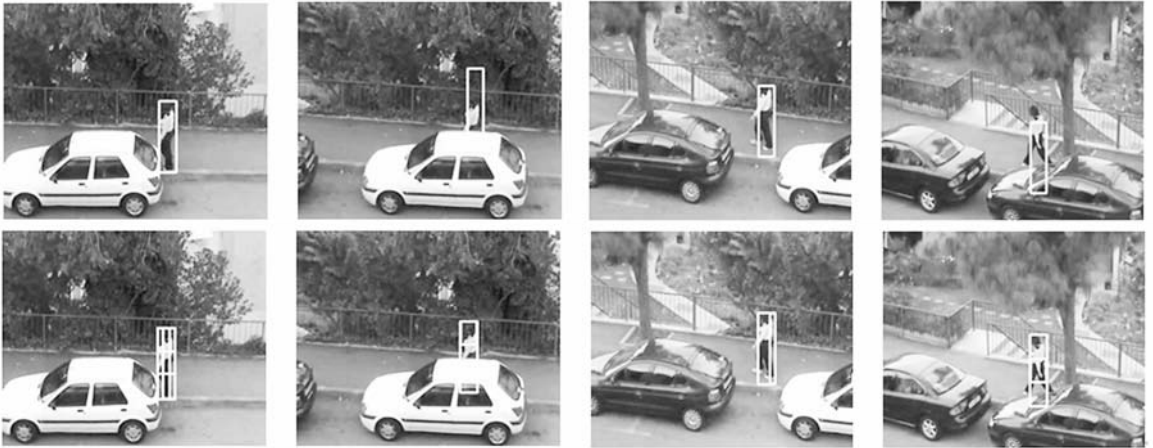


图 4 Woman 视频序列的跟踪结果(帧 1、26、74 和 138。上排:tracker MS 下排:tracker MSF)

Fig. 4 Tracking results of Woman sequence (Frame 1,26,74 and 138. top: tracker MS;bottom: tracker MSF)

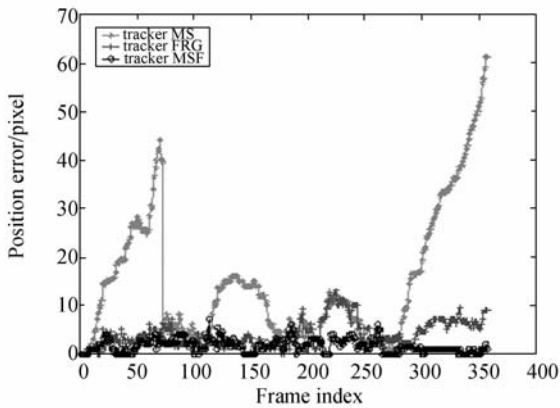


图 5 tracker MS、tracker FRG 和本文提出的 tracker MSF 对 Face 序列的跟踪误差

Fig. 5 Position errors of tracker MS , tracker FRG and tracker MSF for Face sequences

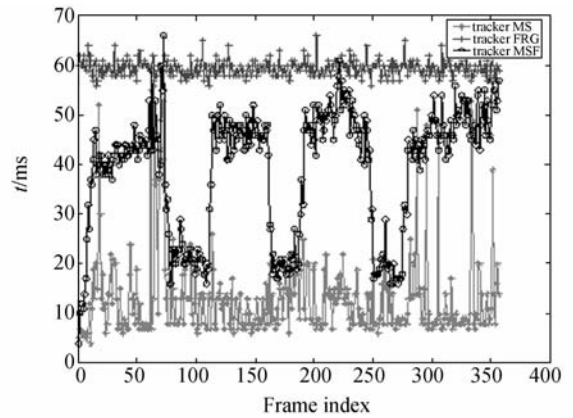


图 6 tracker MS、tracker FRG 和本文提出的 tracker MSF 对 Face 序列的处理时间

Fig. 6 Running time of tracker MS, tracker FRG and tracker MSF for Face sequence

从图 6 可以明显看出 tracker MSF 曲线的起伏,其中波峰处说明目标发生遮挡,而波谷处说明无遮挡。值得注意的是,这种起伏恰好对应于图 5 中的 tracker MS 曲线的起伏(发生遮挡时 tracker MS 的误差明显增大)。当目标未出现遮挡时(如第 78 帧至第 110 帧、第 164 帧至第 189 帧和第 251 帧至第 275 帧处),tracker MSF 计算量仅略高于 tracker MS;而当目标出现遮挡后,由于全部子块都参与 Mean Shift 计算,所以 tracker MSF 的计算量加大。而 tracker FRG 由于采用穷搜索的办法,所以每帧计算量几乎保持不变(平

均处理时间为 60 ms/frame),虽然已采用积分直方图的办法来减小计算量,但仍不能满足实时跟踪的要求。计算可知,前 357 帧 tracker MSF 的平均处理时间为 38.6 ms/frame。而 tracker MSF 对另一段视频(“Woman”)的平均处理时间为 23.2 ms/frame。所以,tracker MSF 在适当的计算量下不仅提高了跟踪性能,而且仍能够满足一般跟踪中 25 frame/s 的要求。

当目标大小发生变化时,可以考虑采取和文献[8]中一样的改进方法,即将跟踪窗口放大和缩小 10%(此时背景选取区域也相应变化)后分别

得到各自的置信度,然后取置信度最佳的跟踪窗口作为最后的目标。当然,这种做法会导致计算量的增加,如何更好地解决此类问题将是以后研究的重点。

6 结 论

本文针对传统 Mean Shift 算法不能很好地对遮挡目标进行跟踪的缺陷,提出将分块思想引入 Mean Shift 框架中的改进办法。首先对跟踪窗口内的目标进行分块,然后在未发生遮挡时仅对外围子块分别实施 Mean Shift 跟踪算法并同时检测遮挡的发生,当遮挡发生后即对所有子块

实施 Mean Shift 跟踪算法。本文使用置信度最高的子块来确定整体目标的最终位置,而子块的置信度由 Bhattacharyya 系数和背景区分度两方面决定。不同视频序列中的目标跟踪结果表明,在发生遮挡时,该算法相比传统 Mean Shift 算法能大大提高跟踪的准确性;相比纯粹的分块算法可大大减少计算量。当目标尺寸为 $70 \text{ pixel} \times 100 \text{ pixel}$ 时,平均处理时间为 38.6 ms/frame ,保证了跟踪的实时性。

目前,本文仅考虑了目标部分遮挡的情况,当目标完全遮挡时和再次出现时如何准确定位目标是需要进一步研究的问题。

参考文献:

- [1] FUKANAGA K, HOSTETLER L. The estimation of the gradient of a density function, with applications in pattern recognition [J]. *IEEE Trans. on Information Theory*, 1975,21(1):32-40.
- [2] 孙中森,孙俊喜,宋建中,等. 一种抗遮挡的运动目标跟踪算法 [J]. *光学精密工程*, 2007,15(2):267-171.
SUN ZH S, SUN J X, SONG J ZH, *et al.*. Anti-occlusion arithmetic for moving object tracking [J]. *Opt. Precision Eng.*, 2007, 15(2): 267-271. (in Chinese)
- [3] YILMAZ A, SHAFIQUE K, SHAH M. Target tracking in airborne forward looking infrared imagery [J]. *Int'l Journal of Image and Vision Computing*, 2003,21(7):623-635.
- [4] COMANICIU D, RAMESH V, MEER P. Kernel-Based object tracking [J]. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 2003,25(5):564-575.
- [5] DEGUCHI K, KAWANAKA O, OKATANI T. Object tracking by the mean-shift of regional color distribution combined with the particle-filter algorithm [M]. United Kingdom: Institute of Electrical and Electronics Engineers Inc, 2004: 506-509.
- [6] COMANICIU D, RAMESH V. Mean shift and optimal prediction for efficient object tracking [M]. *BC: Institute of Electrical and Electronics Engineers Computer Society*, 2000:70-73.
- [7] MAGGIO E, CAVALLARO A. Multi-part target representation for color tracking [C]. *IEEE International Conference on Image Processing*, 2005: 729-732.
- [8] ADAM A, RIVLIN E, SHIMSHON L. Robust fragments-based tracking using the integral histogram [C]. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference*, 2006: 798-805.
- [9] NEJHUM S M S, HO J, YANG M H. Visual tracking with histograms and articulating blocks [C]. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference*, 2008:1-8.
- [10] COLLINS R T, YANX L, LEORDEANU M. Online selection of discriminative tracking features [J]. *IEEE Trans on Pattern Analysis and Machine Intelligence*, 2005,27(10):1631-1643.

作者简介:



颜 佳(1983—),男,湖北天门人,博士研究生,2005年于武汉大学获得学士学位,主要从事目标跟踪、机器视觉等方面的研究。E-mail: yanjiaapple@tom.com



张青林(1983—),男,湖北襄樊人,博士研究生,2005年于武汉大学获得学士学位,主要从事高速图像处理等方面的研究。E-mail: zhang_qinglin83@yahoo.com.cn



吴敏渊(1964—),男,湖北武汉人,副教授,1984年、1989年于武汉测绘科技大学分别获得学士、硕士学位,主要从事图像处理、机器视觉等方面的研究。E-mail: wmy@eis.whu.edu.cn

导师简介:



陈淑珍(1946—),女,湖北武汉人,博士生导师,教授,1970年于武汉大学获得学士学位,主要从事图形图像处理等方面的研究。E-mail: szchen@whu.edu.cn

● 下期预告

裂解连杆螺栓装配机床的动力学仿真及试验

郑黎明,杨慎华,寇淑清,邓春萍

(吉林大学 锻压工艺研究所,吉林 长春 130022)

裂解方法是汽车发动机连杆加工行业的最新技术,定扭矩装配螺栓工序是裂解加工工艺的三大核心工序之一,螺栓拧紧机床的开发对于裂解技术的应用有极大的促进作用。本文介绍了螺栓拧紧机床的基本结构和自动螺栓装配过程,建立了该机床的虚拟样机模型,对装配螺栓过程进行了动态仿真;进行了该机床的扭矩/转角试验,确定了某轿车连杆的螺栓拧紧工艺为 $20 \text{ Nm} + 95^\circ$,在该工艺下,能满足该连杆的螺栓装配要求;最后,讨论了定扭矩装配螺栓工序可能出现的质量缺陷,并提出了相应缺陷的解决方案。仿真结果表明:螺栓能准确穿入到连杆螺栓孔内并装配到连杆上。扭矩/转角试验表明:根据螺栓在装配过程中的伸长变形量可判断螺栓装配的轴向力是否符合要求,进而可确定连杆的定扭矩装配工艺。