棋盘格模板角点的自动识别与定位

王忠石[1),2)] 徐心和[2)]

[1)](沈阳大学信息工程学院，沈阳 110044) [2)](东北大学人工智能与机器人研究所，沈阳 110004)

摘 要 棋盘格模板角点的识别与定位是摄像机定标过程中的关键环节，而自动识别与定位则是实现定标过程自动化的前提条件。为了实现定标过程自动化，提出了一种有效的方法，即利用棋盘格模板图像内部角点的局部灰度特征和由栅格线构成的结构特征，实现了内部角点的自动识别与定位。实验结果表明，该方法是有效和实用的，其能明显地加快定标速度，缩短定标时间，从而为基于多幅棋盘格模板图像的摄像机定标过程自动化创造了条件。

关键词 棋盘格模板 摄像机定标 角点检测 目标识别

中图法分类号：TP242.6+2 TP391.41 **文献标识码**：A **文章编号**：1006-8961(2007)04-0618-05

Auto-recognition and Auto-location of the Checkerboard Pattern Corners

WANG Zhong-shi[1),2)], XU Xin-he[2)]

[1)](*School of Information Engineering, Shenyang University, Shenyang* 110044)

[2)](*Institute of Artificial Intelligence & Robotics, Northeastern University, Shenyang* 110004)

Abstract Recognizing and locating the internal corners of a planar checkerboard pattern image is very important in camera calibration. An effective approach is proposed to automatically recognize and locate the internal target corners of the planar checkerboard pattern image based on the characteristics of the local intensity and the grid line architecture. Experiments show that the approach obviously reduces the time cost for camera calibration, improves the speed of calibration process and is especially adapted for automatic calibration based on multiple images.

Keywords checkerboard pattern, camera calibration, corner detecting, object recognition

1 引 言

棋盘格模板图像是摄像机定标常用的模板图像，而且利用其内部角点与 3 维空间点的对应关系即可实现对摄像机的定标[1,2]。棋盘格模板角点的识别与定位是摄像机定标过程中的关键环节，由于识别与定位的准确程度直接影响定标结果的精度，因此快速准确地识别与定位棋盘格模板角点具有十分重要的意义。

通常所采用的识别与定位方法是人机交互方法，即通过用鼠标按照一定的顺序点击最外边的 4 个角点给出参考信息，然后由计算机通过计算进行识别与定位[3]，但这种方法不仅费时费力，而且难以实现定标过程自动化。

本文利用棋盘格模板图像内部角点的局部灰度特征和由栅格线构成的结构特征，提出了一种有效的识别与定位方法，该方法不仅可以从图像的全部角点中自动地识别与定位所有的目标角点，而且整个过程无任何条件限制，也不需要人的干预，这既能够保证识别的准确性，又能够缩短定标时间，从而为进一步实现定标过程自动化创造了有利条件。

2 棋盘格模板内部角点的特征分析

图 1 为一棋盘格模板图像。每个棋盘格模板内部角点都是 4 个黑白方格的交点。黑白方格构成纵横相交的两组栅格线。这两组栅格线是两组平行线，由透视投影几何知，由于它们在图像上分别汇聚于两个不同的消失点，因此，每个棋盘格模板图像的

收稿日期：2005-09-29；**改回日期**：2006-04-12

第一作者简介：王忠石(1959～)，男，副教授。东北大学博士研究生。主要从事计算机视觉、数字摄影测量等研究。E-mail: wangzhongshi.sd@163.com

内部角点又都是栅格线的交点。综上所述,棋盘格模板图像的内部角点有如下特征:

(1) 角点;

(2) 黑白方格的交点;

(3) 栅格线的交点。

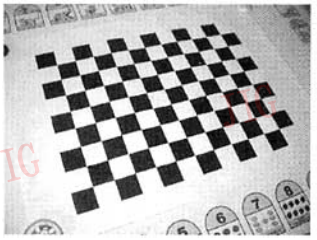

图 1　棋盘格模板图像

Fig. 1　Checkerboard pattern image

3　棋盘格模板内部角点的识别

根据棋盘格模板图像内部角点的特征,可按如下步骤进行识别。

(1) 提取图像中的角点;

(2) 在角点中筛选黑白方格的交点;

(3) 进一步筛选出栅格线的交点。

设棋盘格模板图像为 $\boldsymbol{I}$(如图 1 所示)。采用 harris 角点检测器[4]找出图像中的全部角点,结果如图 2 中标有“+”号的点所示。

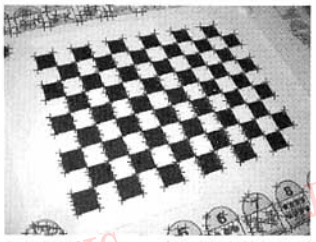

图 2　图像中的全部角点

Fig. 2　All corners in the image

然后对每一个角点,根据黑白方格的灰度构成特点,采用旋转正交模板来筛选黑白方格的交点。正交模板为 2 个 11×11 的模板窗口(如图 3 所示)。该模板中除了在指定位置的值为 1 外,其他位置的值全部为 0。

运算时,先将模板窗口的中心点置于一个角点上,然后在模板窗口范围内,以像素的平均值作为阈值对图像 $\boldsymbol{I}$ 进行分割,得到一个二值图像 $\hat{\boldsymbol{I}}$。

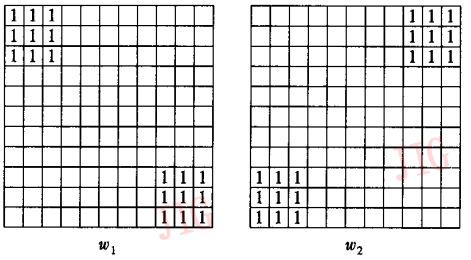

图 3　正交模板

Fig. 3　Rotating orthogonal masks

$$\hat{I}_{i,j}=\begin{cases}1 & w^{I}_{i,j}\geqslant\dfrac{1}{m\times n}\displaystyle\sum_{i=1}^{m}\sum_{j=1}^{n}w^{I}_{i,j}\\ 0 & w^{I}_{i,j}<\dfrac{1}{m\times n}\displaystyle\sum_{i=1}^{m}\sum_{j=1}^{n}w^{I}_{i,j}\end{cases}\tag{1}$$

式(1)中,$w^{I}_{i,j}$ 为图像 $\boldsymbol{I}$ 在正交模板窗口中坐标为(i, j)点的像素值,$m=n=11$ 为窗口的长度和宽度。

在图像 $\boldsymbol{I}$ 中,位于黑方格和白方格中的像素分别具有低灰度值和高灰度值。因此,在图像 $\hat{\boldsymbol{I}}$ 中 0 和 1 分别对应于黑方格和白方格中的像素。因为在每个内部角点周围有两个黑方格和两个白方格,而且黑方格和白方格相对于这个内部角点既对称又正交,所以图像 $\hat{\boldsymbol{I}}$ 中 0 的区域和 1 的区域相对于图像 $\hat{\boldsymbol{I}}$ 的中心点对称且正交。由于在不同的图像中棋盘格模板的方向是不固定的,所以可使用旋转正交模板来搜索黑白方格的交点。

计算图像 $\hat{\boldsymbol{I}}$ 与旋转正交模板 $\boldsymbol{W}$ 相应点乘积的累加和 $\hat{\boldsymbol{I}}*\boldsymbol{W}$,$*$ 为卷积。设两个旋转正交模板分别为 $\boldsymbol{W}_1$ 和 $\boldsymbol{W}_2$,分别用 $\boldsymbol{W}_1$ 和 $\boldsymbol{W}_2$ 做同样的运算,求得两个值 s_1 和 s_2。

$$s_1=\hat{\boldsymbol{I}}*\boldsymbol{W}_1\tag{2}$$

$$s_2=\hat{\boldsymbol{I}}*\boldsymbol{W}_2\tag{3}$$

若 $s_1=0$ 且 $s_2=18$ 或 $s_1=18$ 且 $s_2=0$,则该角点为一个侯选黑白方格的交点。否则,将两个旋转正交模板中的元素顺时针旋转一格,重复相同的运算和判断,直至旋转一周。在此过程中,只要有一次满足 $s_1=0$ 且 $s_2=18$ 或 $s_1=18$ 且 $s_2=0$,则该角点为一个侯选黑白方格的交点。否则,该角点一定不是黑白方格的交点。

对图像中的每一个角点逐一进行处理,在全部角点都处理完后,得到的侯选角点如图 4 所示。在图 4 中,除了棋盘格模板上的内部角点以外,也包括了一些其他的角点,需要进一步处理。

接下来,用栅格线做最后的筛选。为了求出栅格线,先计算出全部侯选角点的质心 $\boldsymbol{p}_c=(x_c,y_c)$。

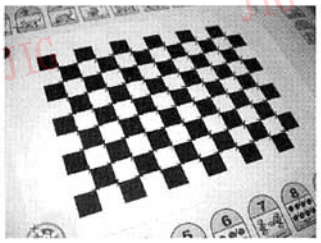

图 4 候选角点

Fig. 4 The candidate corners

$$\begin{cases} x_c = \dfrac{1}{n}\sum_{i=1}^{n} x_i \\ y_c = \dfrac{1}{n}\sum_{i=1}^{n} y_i \end{cases} \tag{4}$$

式(4)中,n 为候选角点数;然后求出距离质心 $\boldsymbol{p}_c=(x_c,y_c)$ 最近并且属于同一方格的 4 个角点 $\boldsymbol{p}_1^c,\boldsymbol{p}_2^c,\boldsymbol{p}_3^c,\boldsymbol{p}_4^c$;最后根据这 4 个角点确定出 4 条靠近质心的直线 L_1^c,L_2^c,L_3^c,L_4^c(如图 5 所示)。

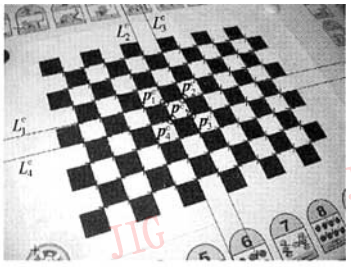

图 5 4 个角点与 4 条栅格线

Fig. 5 The four corners and four grid lines

确定角点时,先分别求出位于这 4 条直线上的角点,并由 L_1^c,L_4^c 上的角点计算出消失点 $\boldsymbol{v}_1$,同样由 L_2^c,L_3^c 上的角点计算出消失点 $\boldsymbol{v}_2$[5,6];然后求出由 $\boldsymbol{v}_1$ 与 L_2^c 上的各个角点所决定的直线,并使一个角点对应一条直线,再分别求出位于每条线上的角点,组成一个角点集 $\boldsymbol{S}^1$[7,8]。对 $\boldsymbol{v}_2$ 与 L_1^c 做同样的运算,组成另一个角点集 $\boldsymbol{S}^2$。

求取全部角点时,先对 $\boldsymbol{S}^1$ 和 $\boldsymbol{S}^2$ 中属于每条线的角点进行交叉筛选,即将 $\boldsymbol{S}^1$ 中属于同一条线的角点构成的第 i 个子集 $\boldsymbol{S}_i^1(i=1,2,\cdots)$ 与 $\boldsymbol{S}^2$ 做交运算

$$\boldsymbol{S} = \boldsymbol{S}_i^1 \cap \boldsymbol{S}^2 \tag{5}$$

若 $\boldsymbol{S}$ 中所包含的角点数大于每条线上角点的平均数,则保留 $\boldsymbol{S}_i$,否则从 $\boldsymbol{S}^1$ 中将 $\boldsymbol{S}_i$ 排除。同理,将 $\boldsymbol{S}^2$ 中由属于同一条线的角点构成的第 i 个子集 $\boldsymbol{S}_j^2(j=1,2,\cdots)$ 与 $\boldsymbol{S}^1$ 做交运算

$$\boldsymbol{S} = \boldsymbol{S}_j^2 \cap \boldsymbol{S}^1 \tag{6}$$

若 $\boldsymbol{S}$ 中所包含的角点数大于每条线上角点的平均数,则保留 $\boldsymbol{S}_j$,否则从 $\boldsymbol{S}^2$ 中将 $\boldsymbol{S}_j$ 排除。

最后,$\boldsymbol{S}=\boldsymbol{S}^1\cap\boldsymbol{S}^2$ 即为所要求的全部角点集,结果如图 6 所示。

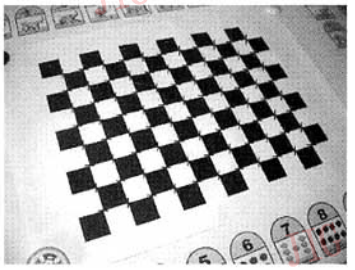

图 6 全部内部角点

Fig. 6 All internal corners

4 棋盘格模板内部角点的定位

棋盘格模板图像内部角点按行列分布,排列成一个有序的阵列,所谓定位就是确定每一个角点在阵列中的位置,即确定其所在的行和列。因为每个内部角点都是栅格线的交点,所以可以利用栅格线和消失点来定位。角点所在的栅格线决定了其所在的行或列,而角点与消失点的距离则决定了其在该行或列中的排列顺序,由此即可将角点的位置完全确定。

4.1 内部角点阵列的确定

设 L_1^c 上的角点为 $\boldsymbol{p}_j^c(j=0,1,\cdots,(n-1))$,则 $\boldsymbol{p}_j^c$ 与消失点 $\boldsymbol{v}_1$ 的距离为

$$D_j = |\boldsymbol{p}_j^c - \boldsymbol{v}_1| \tag{7}$$

将 $\boldsymbol{p}_j^c$ 按 D_j 由小到大排序,使 $\boldsymbol{p}_0^c$ 为离点 $\boldsymbol{v}_1$ 最近的角点,$\boldsymbol{p}_{(n-1)}^c$ 为离点 $\boldsymbol{v}_1$ 最远的角点。

设汇聚于消失点 $\boldsymbol{v}_2$ 的栅格线为 $L_0\sim L_{(n-1)}$(如图 7 所示)。当求出 $L_j(j=0,1,\cdots,n-1)$ 上的全部

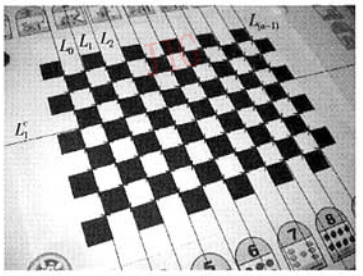

图 7 栅格线

Fig. 7 The grid lines

角点(即位于第 j 列的全部角点)后,$\boldsymbol{p}_{i,j}$($\boldsymbol{p}_{i,j}=(x_{i,j},y_{i,j})$)为 L_j 上的第 i 个角点($i=0,1,\cdots,m-1$),$\boldsymbol{p}_{i,j}$与$\boldsymbol{v}_2$($\boldsymbol{v}_2=(x_{v_2},y_{v_2})$)的距离为

$$D_{i,j}=|\boldsymbol{p}_{i,j}-\boldsymbol{v}_2| \tag{8}$$

则将 $\boldsymbol{L}_j(j=0,1,\cdots,n-1)$上的角点按距离 $D_{i,j}$由小到大进行排序,即可得到所有角点的行列定位。

4.2　棋盘格模板坐标系统参考原点的确定

图像角点的行列坐标与空间模板角点的行列坐标必须相对于相同的参考坐标原点。虽然参考坐标原点的选取可以是任意的,但是必须保持一致才能保证定位的准确。为了实现自动定位,可规定模板的行数和列数奇偶相异,即一个为奇数一个为偶数,并将参考坐标原点设在第 1 个黑方格所在的内部角点上(如图 8 中左上角的圆圈所示)。黑白方格的编号方法为:首先,将模板摆正,使第 1 行和最后一行从左至右第 1 个方格为黑方格,最后一个方格为白方格;然后,按自左至右,由上到下,逐行给黑白方格编号。

按照上述规定,就可利用像素的灰度特征找出第 1 行和最后一行最左边的第 1 个黑方格所在的两个内部角点,再根据这两个角点相对于质心 $\boldsymbol{p}_c$ 的位置关系确定哪一个是参考坐标原点。

设角点的行列定位为

$$\begin{matrix}\boldsymbol{p}_{0,0} & \boldsymbol{p}_{0,1} & \cdots & \boldsymbol{p}_{0,(n-1)}\\ \boldsymbol{p}_{1,0} & \boldsymbol{p}_{1,1} & \cdots & \boldsymbol{p}_{1,(n-1)}\\ \cdots & \cdots & \cdots & \cdots\\ \boldsymbol{p}_{(m-1),0} & \boldsymbol{p}_{(m-1),1} & \cdots & \boldsymbol{p}_{(m-1),(n-1)}\end{matrix}$$

模板 4 个角的方格所在的内部角点是最外侧的两条栅格线上离消失点最近和最远的 4 个角点,即这两条线上的第 1 个角点($\boldsymbol{p}_{0,0}$和$\boldsymbol{p}_{0,(n-1)}$)和最后一个角点($\boldsymbol{p}_{(m-1),0}$和$\boldsymbol{p}_{(m-1),(n-1)}$)。以这 4 个角点附近的局部像素灰度平均值作为阈值 $t_i(i=0,1,2,3)$,对图像进行分割。设每个角点的灰度值为 $I(\boldsymbol{p}_{i,j})$($i=0,1,\cdots,m-1$; $j=0,1,\cdots,n-1$),则这 4 个角点和与其相对角点的中点的灰度值为

$$\begin{cases}I_0=I\left(\dfrac{\boldsymbol{p}_{0,0}+\boldsymbol{p}_{1,1}}{2}\right)\\ I_1=I\left(\dfrac{\boldsymbol{p}_{0,(n-1)}+\boldsymbol{p}_{1,(n-2)}}{2}\right)\\ I_2=I\left(\dfrac{\boldsymbol{p}_{(m-1),0}+\boldsymbol{p}_{(m-2),1}}{2}\right)\\ I_3=I\left(\dfrac{\boldsymbol{p}_{(m-1),(n-1)}+\boldsymbol{p}_{(m-2),(n-2)}}{2}\right)\end{cases} \tag{9}$$

若

$$I_i<t_i,\ (i=0,1,2,3) \tag{10}$$

则相应的方格为黑方格。由此即可确定出两个对应黑方格的角点 $\boldsymbol{p}_0$ 和 $\boldsymbol{p}_1$。

角点 $\boldsymbol{p}_0$ 和 $\boldsymbol{p}_1$ 相对点 $\boldsymbol{p}_c$ 的矢量为

$$\begin{cases}\boldsymbol{v}_0^c=\boldsymbol{p}_0-\boldsymbol{p}_c\\ \boldsymbol{v}_1^c=\boldsymbol{p}_1-\boldsymbol{p}_c\end{cases} \tag{11}$$

令

$$\theta=\varphi_1-\varphi_0 \tag{12}$$

式(12)中 φ_0 和 φ_1 分别为 $\boldsymbol{v}_0^c$ 和 $\boldsymbol{v}_1^c$ 的幅角。若 $0<\theta<\pi$,则参考坐标原点为 $\boldsymbol{p}_0$,否则为 $\boldsymbol{p}_1$。最终定位结果如图 8 所示。

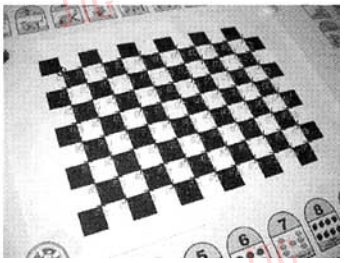

图 8　最终结果图

Fig. 8　The resulting internal corners and the origin point

5　实验结果及分析

为验证本文算法的效果,用多种摄像机所拍摄的棋盘格模板图像对本文算法进行了实验。程序用 MATLAB 6.5 编写,在 Windows 平台下测试通过。程序处理结果是一组内部角点数据,这些数据可以直接被文献[3]中的相应函数用来计算摄像机的内外参数。

为了比较与对照,同时也是为了对所提出算法进行验证,本文以对文献[3]中使用的图像所做的实验为例进行说明。所处理的图像一共 20 幅,可以从下面的网页上下载:http://131.215.134.19/bouguetj/calib_doc/htmls/example.html。图像的分辨率为 640×480pixels。棋盘格模板包含 14×13 个方格,有 156 个内部角点。每个方格的尺寸为 30mm×30mm。

实验时先分别采用文献[3]的人机交互方法和本文的方法来识别与定位棋盘格模板的内部角点;然后应用文献[3]中的相应函数来计算摄像机的内外参数。定标结果和定标误差如表 1 和表 2 所示。

对这 20 幅图像的角点进行识别与定位所用的时间以及定标全过程所用的时间如表 3 所示。

表 1 定标结果

Tab. 1 Results of calibration

	定标结果
交互方法	f = [657.462 90;657.946 73]; c = [303.136 65;242.569 35]; k = [-0.254 03;0.121 43; -0.000 21;0.000 02;0.000 00]
本文方法	f = [657.462 37;657.946 04]; c = [303.137 42;242.569 93]; k = [-0.254 03;0.121 43; -0.000 21;0.000 02;0.000 00]

表 2 定标误差

Tab. 2 Errors of calibration

	定标误差(pixel)
交互方法	e_f = [0.318 19;0.340 46]; e_c = [0.646 82;0.592 18]; e_k = [0.002 48;0.009 86;0.000 13;0.000 13;0.000 00]
本文方法	e_f = [0.318 17;0.340 44]; e_c = [0.646 78;0.592 15]; e_k = [0.002 48;0.009 86;0.000 13;0.000 13;0.000 00]

表 3 定标时间

Tab. 3 Time

	定标时间(s)	
	交互方法	本文方法
提取角点	1 064.54	192.21
全部过程	1 572.06	253.38

由表 1 和表 2 中可以看出,两种不同的方法所获得的结果是近乎一致的。在定标过程中,除了识别与定位棋盘格模板内部角点所用的程序以外,两种方法所调用的程序函数几乎完全相同,这也验证了本文所提出的识别与定位棋盘格模板内部角点的方法是有效的。表中 f、c 和 k 分别表示焦距、主点和畸变系数,e_f、e_c 和 e_k 为相应量的误差[3]。

由表 3 可以看出,在识别与定位棋盘格模板内部角点所用的时间上,本文的方法所用时间远远小于交互式的方法,而整个定标过程所节省的时间则更为突出。这主要是因为使用交互式的方法时,为了获取同样高的精度,要对重投影误差进行人工分析,并对误差较大的图像的角点做细致调整所致。而且这个过程也是比较费时和费力的。而采用本文的方法,角点的识别与定位和定标参数计算则是一次性地连续完成的。这不仅节省了时间,而且避免了令人厌倦的重复性工作。因为参数计算所用的时间很少,所以整个定标过程所节省的时间应主要地归功于角点识别与定位过程。

本文所提出的方法由于能够不间断地连续对多幅棋盘格模板图像的内部角点进行识别与定位,从而为定标过程自动化创造了十分有利的条件。

6 结 论

棋盘格模板角点的识别与定位是摄像机定标过程中的关键环节,而自动识别与定位则是实现定标过程自动化的前提条件。本文利用棋盘格模板图像内部角点的局部灰度特征和由栅格线构成的结构特征,实现了内部角点的自动识别与定位。该方法不仅能够自动准确地识别与定位内部角点,而且不需要人做任何干预,从而明显地提高了工作效率。实验结果表明,该方法是有效和实用的,其能明显地加快定标速度,缩短定标时间,为基于多幅图像的摄像机定标过程自动化创造了条件。

参考文献(References)

1 Zhang Z. A flexible new technique for camera calibration[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2000, **22**(11): 1330 ~ 1334.

2 Heikkilä J, Silvén O. A four-step camera calibration procedure with implicit image correction [A]. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition(CVPR'97)[C], San Juan, Puerto Rico, 1997: 1106 ~ 1112.

3 Bouguet Jean-Yves. Camera Calibration Toolbox for Matlab [EB/OL]. http://131.215.134.19/bouguetj/calib_doc/index.html.

4 Harris C, Stephens M. A combined corner and edge detector[A]. In: Proceedings Alvey Vision Conference [C], Manchester, UK, 1988: 189 ~ 192.

5 McLean G, Kotturi D. Vanishing point detection by line clustering [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1995, **17**(11): 1090 ~ 1095.

6 Kosecka J, Zhang W. Efficient computation of vanishing points[A]. In: Proceedings of The 2002 IEEE International Conference on Robotics and Automation(ICRA'02)[C], Washington DC, USA, 2002: 3321 ~ 3327.

7 Faugeras O. Stratification of 3-D vision: projective, affine, and metric representations[J]. Journal of the Optical Society of America, 1995, **12**(3):465 ~ 484.

8 Burns J B, Hanson A R, Riseman E M. Extracting straight lines[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1986, **8**(4): 425 ~ 455.