文章编号：1007-2780(2022)07-0913-11

基于改进热图损失函数的目标6D姿态估计算法

林　林[1,2]，王延杰[1*]，孙海超[1]
(1. 中国科学院 长春光学精密机械与物理研究所，吉林 长春 130033；
2. 中国科学院大学，北京 100049)

摘要：针对传统热图回归使用的均方误差(MSE)损失函数训练热图回归网络的精度不高且训练缓慢的问题，本文提出了用于热图回归的损失函数Heatmap Wing Loss(HWing Loss)。该损失函数对于不同的像素值有不同的损失函数值，前景像素的损失函数梯度更大，可以使网络更加关注前景像素，使热图回归更加准确快速。同时根据热图分布特性，使用基于高斯分布的关键点推理方法减小热图推断关键点时的量化误差。以此两点为基础，构造新的基于关键点定位的单目标姿态估计的算法。实验结果表明，相比于使用MSE Loss的算法，使用HWing Loss的姿态估计算法有更高的ADD(-S)准确率，在LINEMOD数据集上达到了88.8%，性能优于近期其他的基于深度学习的姿态估计算法。本文算法在RTX3080 GPU上最快能以25 fps的速度运行，兼具速度与性能优势。

关　键　词：深度学习；姿态估计；损失函数；热图
中图分类号：TP391.4　　**文献标识码**：A　　**doi**：10.37188/CJLCD.2021-0317

Object 6D pose estimation algorithm based on improved heatmap loss function

LIN Lin[1,2], WANG Yan-jie[1*], SUN Hai-chao[1]

(1. *Changchun Institute of Optics, Fine Mechanics and Physics, Chinese Academy of Sciences, Changchun* 130033, *China*;
2. *University of Chinese Academy of Sciences, Beijing* 100049, *China*)

Abstract: In view of the problem of low precision and slow training of heatmap regression network trained by mean square error (MSE) loss function used in traditional heatmap regression, the loss function Heatmap Wing Loss (HWing Loss) for heatmap regression is proposed in this thesis. In terms of different pixel values, the loss function has different loss function values, and the loss function gradient of foreground pixels is larger, which can make the network focus more on the foreground pixels and make the heatmap regression more accurate and faster. In line with the distribution characteristics of the heatmap, the keypoint inference method based on the Gaussian distribution is adopted in this thesis to reduce the quantization error when the heatmap infers the keypoints. By taking the two points as the basis, it constructs a new monocular pose estimation algorithm based on keypoint positioning. According to the

收稿日期：2021-12-03；**修订日期**：2022-01-09.
基金项目：吉林省科技发展计划(No. 20210201132GX)
Supported by Department of Science and Technology of Jilin Province (No. 20210201132GX)
*通信联系人，E-mail: wangyj@ciomp.ac.cn

experiments, in contrast with the algorithm using MSE Loss, the pose estimation algorithm using HWing Loss has a higher ADD(-S) accuracy rate, which reaches 88.8% on the LINEMOD dataset. Meanwhile, the performance is better than other recent pose estimation algorithms based on deep learning. The algorithm in this thesis can run at the fastest speed of 25 fps on RTX3080 GPU, in which the high speed and performance can be both embodied.

Key words: deep learning; pose estimation; loss function; heatmap

1 引 言

随着计算机视觉技术的不断发展,仅从图像中检测物体已经不能满足智能机器人对环境感知的需要,基于图像的物体6D姿态测量成为了新的研究热点。用RGB-D图像估计物体的姿态虽然精度高,但是计算复杂难以应用。使用单RGB图像配合目标的三维模型可以解决单目相机成像时的尺度不确定问题,也可以得到较好的结果,但在实际应用场景下,目标的背景较为复杂,而且会有很多遮挡现象出现,这都给姿态估计问题造成了很大的影响。因此,如何从单张图像中快速准确地在复杂场景下获得目标的6D姿态是一个非常值得深入研究的课题。

传统姿态估计算法通常使用手工特征[1-3]建立图像模板与图像之间的对应关系,很难处理无纹理对象,速度也较慢。随着深度学习的发展,基于CNN的姿态估计算法取得了优异的性能。Deep-6DPose[4]、AAE[5]等端到端算法是将图像输入到神经网络中直接输出目标姿态,但是这种方法的泛化能力并不好,网络无法学习到足够的特征以表达目标姿态。近期研究的两阶段算法如yolo-6d[6]、BetaPose[7]、Pix2Pose[8]、PVNet[9]等在精度上相比于端到端算法展现了较大的优势。其中BetaPose[7]为这类方法提供了一个简单高效的基础架构,即先使用神经网络回归热图定位关键点,再使用P*n*P算法计算目标姿态。

热图是一种特殊的图像,图像上每个像素值代表关键点出现在该位置的概率值,可以用于神经网络回归关键点时的中间表示,利用预测得到的热图极大值及其邻域推断得到关键点位置。因此,热图的前景像素的预测精度对于关键点定位任务至关重要,这个部分即使出现了很小的预测误差,也会导致很大的关键点偏离。相反,预测热图的背景像素值并不重要,预测时只需使其逐步趋向于零值即可。而BetaPose并没有利用好热图的相关特性:首先,在回归热图时,使用传统的MSE Loss损失函数。MSE Loss对于不同位置的像素值没有区分,训练被大量无意义的背景像素主导,导致回归的热图精度不高。其次,在使用预测的热图定位关键点时,只简单地取最大值点,存在量化误差。这两点影响了算法最终的结果。

本文对BetaPose进行了改进。首先针对MSE Loss损失函数对像素没有区分的问题,提出适用于热图回归的损失函数Heatmap Wing Loss(HWing Loss),该损失函数更加关注前景像素误差,可以有效提高热图回归的质量。其次,使用基于高斯分布的关键点推理方法,减小热图推断关键点时的量化误差。以此两点为基础,构造了新的单目标姿态估计算法。实验结果表明,相比于其他姿态估计算法,本文算法在LINEMOD数据集上有更高的ADD(-S)准确率。

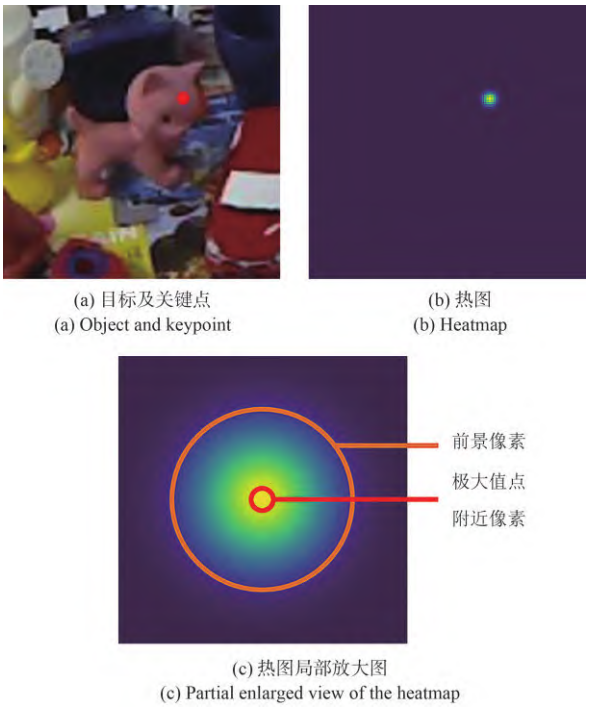
2 适用于热图回归的损失函数

2.1 热图以及传统损失函数

如前文所述,热图绝大多数像素都是背景像素,对关键点至关重要的像素非常少,如图1所示,其中图1(a)为目标图像,红色部分为目标的关键点,图1(b)为该关键点对应的热图,图1(c)为图1(b)热图的局部放大图。因此,我们希望用合适的损失函数使网络训练可以对不同像素进行区分,使前景像素训练收敛速度更快,同时要对前景像素的小误差有更高的敏感度。

在热图回归中,常使用损失函数MSE Loss即均方误差损失函数。但MSE Loss梯度随着误差增大而增大,大误差的损失梯度要高于小误差,这导致神经网络更关注于大误差部分,而忽略小误差部分。

Adaptive Wing Loss[10](AWing Loss)提高了



(a) 目标及关键点
(a) Object and keypoint

(b) 热图
(b) Heatmap

(c) 热图局部放大图
(c) Partial enlarged view of the heatmap

图 1　关键点热图

Fig. 1　Keypoint and heatmap

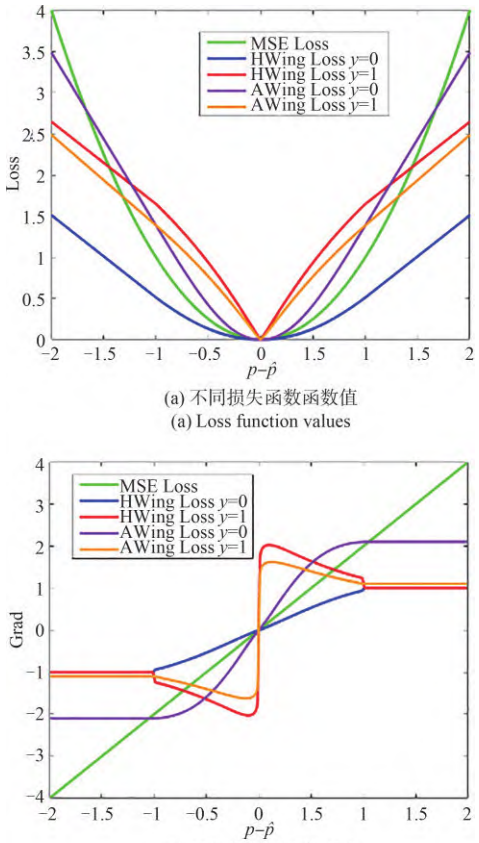
小误差的敏感度，并对前景像素与背景像素做了区分。但AWing Loss前景像素梯度只在误差为0值附近大于背景像素，这导致在网络训练的大部分时期，网络都是向着背景像素误差减小方向前进，与热图前景像素更重要的特性背道而驰。MSE Loss和AWing Loss的表达式如式(1)、(2)所示。

$$\text{MSE Loss}(p,\hat{p})=|p-\hat{p}|^2, \tag{1}$$

$$\text{AWing Loss}(p,\hat{p})=\begin{cases}\omega\ln\left(1+|p-\hat{p}|^{\varepsilon-p}\right) & |p-\hat{p}|<\theta\\ A|p-\hat{p}|-C & |p-\hat{p}|>\theta\end{cases}. \tag{2}$$

2.2　Heatmap Wing Loss

根据热图本身特性，理想的热图损失函数应该有以下特性：(1)大误差部分具有恒定梯度，不会导致产生梯度爆炸的问题。(2)背景像素的小误差部分无需精准的预测，只需要逐步回归到零值，损失函数的特性与MSE Loss损失函数类似。(3)前景图像的小误差部分的回归精确程度是热图回归的关键，对小误差要有更大的梯度。损失函数特性与AWing Loss函数类似。并且在整个训练过程中前景像素的梯度都应该大于背景像素的梯度，这也有利于网络向着重要的前景像素损失降低方向训练。本文在AWing Loss的基础上提出适用于热图回归的损失函数Heatmap Wing Loss(HWing Loss)，其表达式如公式(3)所示。



(a) 不同损失函数函数值
(a) Loss function values

(b) 不同损失函数梯度值
(b) Loss function gradient values

图 2　不同损失函数对比图

Fig. 2　Diagram of loss functions

$$\text{HWing Loss}(p,\hat{p})=\begin{cases}\omega\ln\left(1+\left|\dfrac{p-\hat{p}}{\delta-p}\right|^{\varepsilon-p}\right) & |p-\hat{p}|<\theta\\ |p-\hat{p}|-C & |p-\hat{p}|>\theta\end{cases}. \tag{3}$$

HWing Loss采用分段函数设计，p与$\hat{p}$分别代表真实像素值与预测像素值，$p-\hat{p}$为图像像素误差。θ为阈值，在0～1之间。δ、ε、ω为参数，其中$\delta>1$，$\varepsilon>2$。

为了提升前景像素损失函数的梯度，让整个网络训练过程能以前景像素主导，HWing Loss相比于AWing Loss做了两点改进。

首先是小误差部分，对像素误差$p-\hat{p}$增加

系数 $1/(\delta-p)$，对误差进行缩放。此时将 AWing Loss 与 HWing Loss 函数看作误差 $p-\hat{p}$ 的函数。在 $p-\hat{p}>0$ 部分，其梯度分别如公式(4)、(5)所示。可以看出，当参数 $\delta\in(1,2)$，$\varepsilon>2$ 时，对于背景像素 $p=0$，公式(5)中 $(\delta-p)^{\varepsilon-p}>1$，此时 HWing Loss 的函数梯度小于 AWing Loss，即 HWing Loss 中 $1/(\delta-p)$ 部分减小了损失函数的梯度。而对前景像素 $p=1$，公式(5)中 $(\delta-p)^{\varepsilon-p}<1$，HWing Loss 的函数梯度大于 AWing Loss，即 HWing Loss 中 $1/(\delta-p)$ 部分放大了损失函数的梯度。因此，HWing Loss 中增加的系数 $1/(\delta-p)$ 可以在前景像素处对损失函数梯度进行放大，同时缩小背景像素的损失函数，从而使损失函数在前景像素的梯度大于背景像素。其次，对大误差部分，取消 AWing Loss 中的系数 A，从而避免在大误差部分背景像素梯度值大于前景像素的问题。

$$\frac{d}{d(p-\hat{p})}\mathrm{AWing\ Loss}(p,\hat{p})=\omega\frac{(\varepsilon-p)(p-\hat{p})^{\varepsilon-p-1}}{1+(p-\hat{p})^{\varepsilon-p}},\tag{4}$$

$$\frac{d}{d(p-\hat{p})}\mathrm{HWing\ Loss}(p,\hat{p})=\omega\frac{(\varepsilon-p)(p-\hat{p})^{\varepsilon-p-1}}{(\delta-p)^{\varepsilon-p}+(p-\hat{p})^{\varepsilon-p}}.\tag{5}$$

图 2 展示了 HWing Loss 分别在背景像素 $p=0$(蓝色图像)和前景像素 $p=1$(红色图像)时与 MSE Loss(绿色图像)的比较，其中参数 $\delta=1.8$，$\varepsilon=2.1$。可以看出，HWing Loss 在各点都是连续的，且梯度在 0 处连续。在 $p=1$ 即前景像素部分，其特性与 AWing Loss 函数相像；在 $p=0$ 即背景像素部分，其特性类似于 MSE Loss。由图 3(b)可以看出，本文提出的 HWing Loss 在小误差部分，误差 $p-\hat{p}$ 一定的情况下，损失函数梯度会随着像素 p 增大而增大，大误差部分的损失函数梯度保持梯度恒定。这符合上文提到的理想损失函数特性，从而可以使整个网络训练过程都以前景像素为主。

使用 HWing Loss 损失函数训练热图回归网络时，训练开始所有像素都有较大误差，都应用损失函数的大误差部分，同时减小误差。当误差减小至 θ 后，背景像素随着误差减小，梯度逐渐减小，使得网络逐渐不再关注其收敛。而前景像素随着误差减小，梯度逐渐增大，在网络损失中占据主导位置，即网络会继续向着前景像素误差减小的方向训练，这符合热图前景像素更加重要的特点。

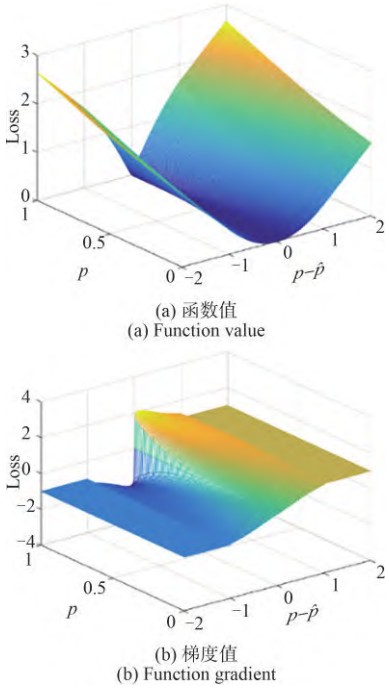


图 3 HWing Loss 示意图

Fig. 3 Diagram of HWing Loss

3 目标姿态估计算法

本文姿态估计算法为两阶段算法：第一阶段预测热图，用于定位目标关键点；第二阶段由关键点计算目标姿态。具体流程如图 4 所示。首先将输入图像裁剪为 $H\times W$ 大小，输入热图回归网

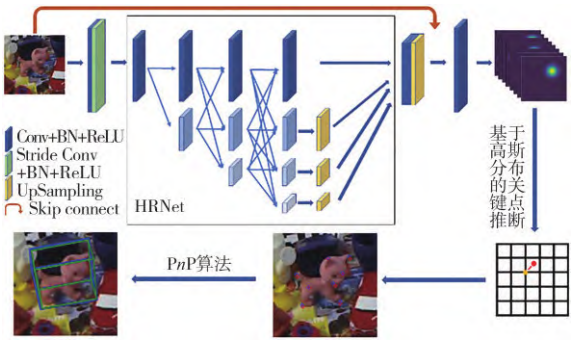


图 4 姿态估计算法流程图

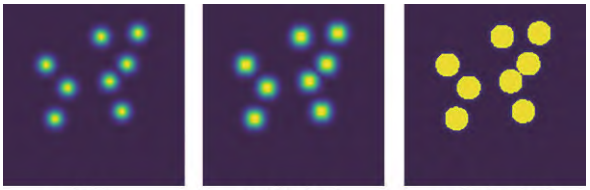
Fig. 4 Flowchart of the pose estimation algorithm

络，得到输出的热图，由预测热图推理关键点位置，最后使用PnP算法计算得到姿态。

3.1　热图回归网络

本文使用HRNet[11-12]作为骨干网络构造热图回归网络。为减小网络运算量，使用步长为2的卷积将特征图由$H\times W$降至$H/2\times W/2$，输入到HRNet中。而为了减小特征网络输出计算关键点位置时，由于特征图与原图尺度不一致需要缩放产生的误差，将HRNet输出的特征进行上采样至原图大小，并与原图进行拼接，经过最后一个卷积块的运算，得到原图大小的热图。我们使用$H\times W\times 3$大小的图像作为输入，每张图像定义C个关键点时，网络的输出为$H\times W\times C$的热图，在训练时使用上文提出的HWing Loss进行像素级的监督。

由于热图的前景像素部分只占整个图像的不足1%，即使我们使用了HWing Loss提高网络对前景像素的关注度，在训练时，整个损失仍然会以重要性不高的背景像素为主，导致网络不能更精确地得到前景像素的预测。因此，我们使用文献[10]中的策略，在计算网络损失时，对前景像素与背景像素给予不同的权重。如图5所示，首先将热图进行灰度膨胀操作，在训练损失计算时，对膨胀热图像素值高于0.2的部分给予10倍的权重，使网络更加关注于前景像素的误差。



(a) 热图 (a) Heatmaps　(b) 膨胀热图 (b) Heatmaps after dilation　(c) 损失加权图 (c) Weighted loss map

图5　加权损失示意图

Fig. 5　Schematic diagram of weighted loss

3.2　关键点推断

在传统方法中，通常使用热图的最大值位置作为关键点的预测值。但是由于图像是离散的，而关键点的位置很有可能不在像素点位置，这就导致了只用最大值位置作为关键点预测值会产生量化误差，影响最终姿态估计结果。

Zhang等[13]提出了基于高斯分布的热图推理关键点的方法，但在二维图像上进行运算耗时较多。本文对其进行一维简化。

考虑到热图是由二维高斯函数生成，根据高斯函数的可分离性，二维高斯函数可以分解为两个一维高斯函数。

$$G(x,y)=\frac{1}{2\pi\sigma^2}\mathrm{e}^{-\frac{x^2+y^2}{2\sigma^2}}=\frac{1}{\sqrt{2\pi}\,\sigma}\mathrm{e}^{-\frac{x^2}{2\sigma^2}}\times\frac{1}{\sqrt{2\pi}\,\sigma}\mathrm{e}^{-\frac{y^2}{2\sigma^2}},\tag{6}$$

极值点在u处的一维高斯函数为：

$$G(x)=\frac{1}{\sqrt{2\pi}\,\sigma}\mathrm{e}^{-\frac{(x-u)^2}{2\sigma^2}}.\tag{7}$$

为降低逼近难度，对高斯函数进行对数运算，将高斯函数变为二次函数，同时，可以保持极值点位置。

$$L(x)=-\frac{(x-u)^2}{2\sigma^2}-\frac{1}{2}\ln 2\pi-\ln\sigma,\tag{8}$$

其中在极值点u处，$L(x)$以及其一阶二阶导数值分别为：

$$L(u)=L(x)\mid_{x=u}=-\frac{1}{2}\ln 2\pi-\ln\sigma,\tag{9}$$

$$L'(u)=L'(x)\mid_{x=u}=-\frac{x-u}{\sigma^2}\mid_{x=u}=0,\tag{10}$$

$$L''(x)\equiv-\frac{1}{\sigma^2}.\tag{11}$$

假设预测热图极大值在x方向坐标值为u_0，易知u_0位于极值点u附近，可以通过u_0推断u位置。我们对$L(x)$在u_0位置进行泰勒展开，忽略三阶小量：

$$L(u)\approx L(u_0)+L'(u_0)(u-u_0)+L''(u_0)\frac{(u-u_0)^2}{2}.\tag{12}$$

联立式(9)、(11)、(12)可得：

$$u=u_0-\frac{L'(u_0)}{L''(u_0)},\tag{13}$$

其中，$L'(u_0)$、$L''(u_0)$分别可由预测热图在最大值位置在x方向上的一阶差分与二阶差分计算得到。

同理，对y方向坐标有：

$$v=v_0-\frac{L'(v_0)}{L''(v_0)}.\tag{14}$$

我们使用公式(13)、(14)进行关键点优化。

3.3　姿态计算

通过预测热图推得关键点位置时，由于目标

可能被遮挡,不可见的关键点会出现较大的定位误差从而影响姿态估计,所以需要对关键点进行筛选。考虑到热图每个像素值的本质是该关键点出现在该位置的概率,可以使用预测热图极值位置的像素值作为该关键点的置信系数,并对关键点进行筛选。为了保证姿态计算的稳定性,选择置信系数大的K个图像关键点。

获取图像关键点后,结合已知的三维关键点以及相机的内参,可以通过求解PnP问题计算得到姿态。我们使用EPnP算法[14]进行PnP问题的求解。

4 实验与结果

4.1 数据集

实验中使用的数据集包括LINEMOD数据集和Occlusion LINEMOD数据集。

LINEMOD数据集是6D目标姿态估计的标准数据集,这个数据集包含多个姿态估计场景,比如复杂背景、无纹理目标的场景。每一幅图像的中心都有一个标记了平移、旋转和类别的目标,同时该数据集还提供了每个目标的3D模型。在LINEMOD数据集中共有15 783张图像和13类目标,每个目标特征大约有1 200个实例。

Occlusion LINEMOD数据集是对LINEMOD数据集的扩展,每一幅图像包含一个被标记的目标,大部分图像中的目标被部分遮挡。在实验中,Occlusion LINEMOD数据集仅用来测试,只用LINEMOD数据集进行训练。

4.2 性能评估

我们使用ADD(-S)指标评估算法性能,其中ADD指标度量的是模型顶点之间3D平均距离。如果3D模型顶点的坐标与估计的坐标之间的平均距离小于目标直径的10%,预测就是正确的。对于对称对象,使用ADD-S指标度量,其平均距离是基于最近的点距离计算的。性能评估时以测试集中预测正确的图像数量与总数的百分比数计算。

4.3 实验参数

本文使用PyTorch搭建训练环境。在LINEMOD数据集中每个类别随机选择30%的图像作为训练集,其余70%作为测试集。为了防止过拟合及增大训练空间,我们在训练集内添加合成图像。对于每个对象,我们渲染了10 000张视点均匀采样的图像。同时用剪切和粘贴策略合成了另外30 000张图像,每幅合成图像的背景随机采样自SUN397[15]。

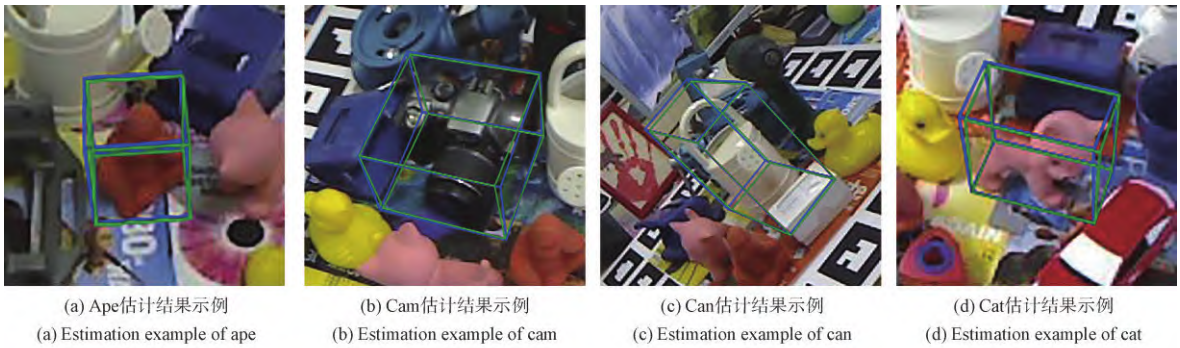
我们在目标三维模型上使用迭代最远点算法标注32个关键点并创建监督热图。输入图像裁剪至128×128,每一批次训练32张图像,学习率设置为0.01,训练30个轮次。对于损失函数HWing Loss,经过多次实验,选择参数$\omega=14,\theta=0.5,\delta=1.8,\varepsilon=2.1$。在测试过程中,选择置信系数最大的24个关键点输入EPnP算法进行姿态估计。

4.4 单目标姿态估计

我们在LINEMOD数据集上进行了单目标姿态估计测试,部分结果如图6所示。可以看出,无论是复杂背景还是无纹理目标,本文算法都有出色的姿态估计结果。

为验证本文算法的有效性,我们将本文算法与近期算法进行对比。作为对比的BetaPose[7]、yolo-6d[6]、PVNet[9]是两阶段算法,区别在于关键点间接表示形式。BetaPose[7]与本文相同,使用热图,yolo-6d[6]直接回归关键点坐标。PVNet[9]使用图像像素点指向关键点的方向向量作为关键点的间接表达方式,该算法基于目标分割,在网络回归时需要增加单独的图像分割分支。PoseCNN[16]是端到端算法,可以直接从图像中计算得到姿态。DeepIM[17]在PoseCNN后增加了迭代优化的后处理部分。ADD(-S)指标测试结果如表1所示,其中eggbox和glue为对称对象,使用ADD(-S)度量,其余对象使用ADD度量。加粗数字为本类别中准确率最高的结果。

由测试结果可以看出,本文算法在无需后优化算法(细化算法,在粗姿态的基础上继续迭代优化)中有着最高的平均准确率,并且在13个类别中的8个类别准确率领先于其他算法。相比于同样使用热图回归关键点的BetaPose,本文算法的准确率提升了16.2%,主要原因在于我们使用的热图回归损失函数HWing Loss可以提升热图回归的质量。本文算法相比于PVNet的平均准确率虽然只领先2.5%,但是PVNet需要额外分割图像,训练收敛困难。并且PVNet使用像素级的方向向量作为关键点的间接表达方式,关键点



(a) Ape估计结果示例 (a) Estimation example of ape　(b) Cam估计结果示例 (b) Estimation example of cam　(c) Can估计结果示例 (c) Estimation example of can　(d) Cat估计结果示例 (d) Estimation example of cat

(e) Driller估计结果示例 (e) Estimation example of driller　(f) Duck估计结果示例 (f) Estimation example of duck　(g) Eggbox估计结果示例 (g) Estimation example of eggbox　(h) Glue估计结果示例 (h) Estimation example of glue

图6　部分单目标姿态估计结果(绿色框为真实姿态的目标三维边界框，蓝色框为估计姿态的目标三维边界框)

Fig. 6　Qualitative results for single object pose estimation(green 3D bounding boxes represent the ground truth poses, and blue 3D bounding boxes represent our predictions)

表1　单目标姿态估计ADD(-S)指标测试结果

Tab. 1　Single object pose estimation ADD(-S) metric test results

算法	Ours	[7]	[6]	[9]	[16]	[16-17]
ape	**64.8**	41.2	21.6	43.6	—	**77.0**
bench	99.2	85.7	81.8	**99.9**	—	97.5
cam	**88.3**	78.9	36.6	86.9	—	**93.5**
can	**96.8**	85.2	68.8	95.5	—	96.5
cat	**82.7**	73.9	41.8	79.3	—	82.1
driller	**98.0**	77.0	63.5	96.4	—	95.0
duck	**60.1**	42.7	27.2	52.6	—	**77.7**
eggbox*	98.9	78.9	69.6	**99.2**	—	97.1
glue*	**99.1**	72.5	80.0	95.7	—	**99.4**
hole	78.4	63.9	42.6	**81.9**	—	52.8
iron	96.8	94.4	75.0	**98.9**	—	98.3
lamp	97.3	98.1	71.1	**99.3**	—	97.5
phone	**93.6**	51.0	47.7	92.4	—	87.7
平均值	**88.8**	72.6	56.0	86.3	62.7	88.6

*对称对象

推理阶段计算复杂度较高。同时，本文算法的准确率相比于PoseCNN提升了26.1%；而PoseCNN使用DeepIM优化后，算法准确率仍然不及本文算法。可以说本文算法有非常优秀的单目标姿态估计准确性。

表2给出了本文算法在LineMod数据集的不同类别下的目标直径d_{obj}，关键点定位误差e_p以及姿态估计误差e_t、e_R，其计算公式如式(15)～(17)所示。

$$e_p=\frac{1}{N}\sum_N\left\| p-\hat{p}\right\|_2, \tag{15}$$

$$e_t=\frac{1}{N}\sum_N\left\| \boldsymbol{t}-\hat{\boldsymbol{t}}\right\|_2, \tag{16}$$

$$e_R=\frac{1}{N}\sum_N\arccos\left(\frac{\mathrm{Tr}\left(\boldsymbol{R}\hat{\boldsymbol{R}}^{-1}\right)-1}{2}\right), \tag{17}$$

其中p、$\boldsymbol{t}$、$\boldsymbol{R}$分别代表关键点定位、姿态平移向量、姿态旋转矩阵的测量值，$\hat{p}$、$\hat{\boldsymbol{t}}$、$\hat{\boldsymbol{R}}$则分别代表关键点定位、姿态平移向量、姿态旋转矩阵的真实值。从表2中可看出在多数类别下，算法在13个类别的平均关键点定位误差为3.02个像素。不同类别的关键点定位误差主要取决于目标尺寸而与目标形状纹理关系不大，这体现了本文算法在复杂背景、无纹理目标的场景下优秀的关键点定位能力。对于类别内关键点，其定位误差与三维关键点与目标中心的距离相关。

以cat类别为例，图7展示了cat类别的32个

表 2 关键点定位误差与姿态估计误差

Tab. 2 Keypoint positioning error and pose estimation error

	d_{obj}/cm	e_p/pixel	e_t/cm	e_R/(°)
ape	9.74	1.47	0.95	2.76
bench	28.69	2.90	0.87	1.84
cam	17.15	2.02	0.84	1.97
can	19.34	2.84	0.80	1.32
cat	15.26	2.11	0.90	2.33
duck	25.94	1.78	1.08	2.76
driller	10.71	4.72	0.87	1.76
eggbox*	17.63	3.93	1.35	1.69
glue*	16.48	2.23	1.25	2.58
hole	14.82	2.29	1.00	2.37
iron	30.31	5.17	1.13	2.37
lamp	28.55	5.12	1.24	2.19
phone	20.83	2.74	0.92	2.18
平均值	/	3.02	1.02	2.16

*对称对象

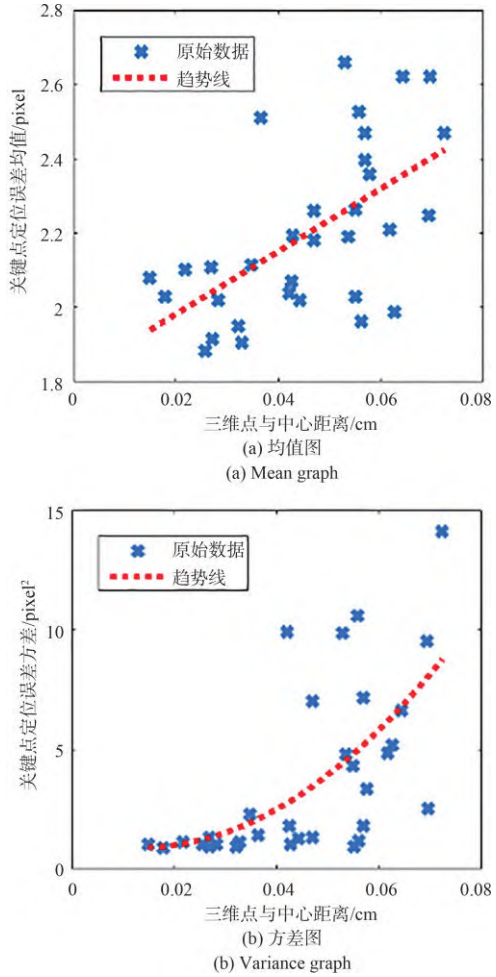


图 7 图像关键点误差与三维关键点偏离中心距离关系

Fig. 7 Relationship between the keypoint error and the distance between the 3D keypoint and the object center

图像关键点的定位误差均值、方差与三维关键点偏离目标中心的关系。可以看出随着三维关键点偏离目标中心，图像关键点定位误差的均值与方差都趋向于增大。但是三维更加分散的关键点选取有利于提升 P*n*P 算法计算姿态时的鲁棒性，因此从整体上看，由于三维关键点偏离目标中心导致的图像关键点定位误差增大是可以接受的。

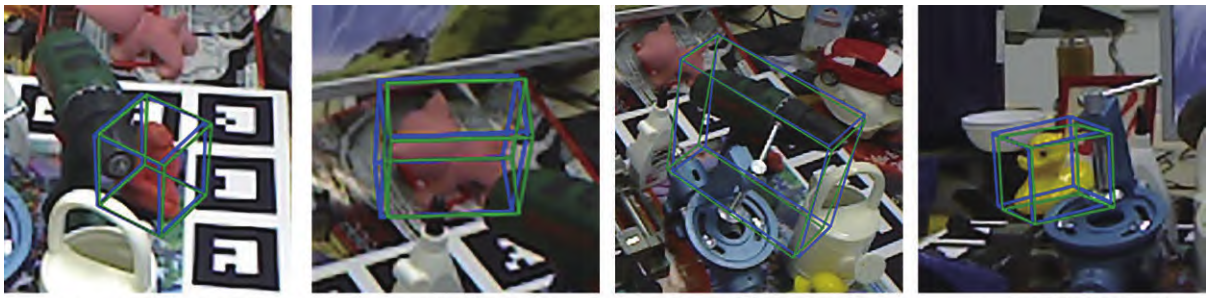
在姿态估计误差上，算法在 13 个类别的平均平移误差为 1.02 cm，平均旋转误差为 2.16°。姿态估计误差主要来源于关键点误差，包括关键点的绝对定位误差与关键点中的离群点。其次，数据集中的相机内参与姿态真值的不准确同样会影响最后测量的准确性。

4.5 遮挡目标姿态估计

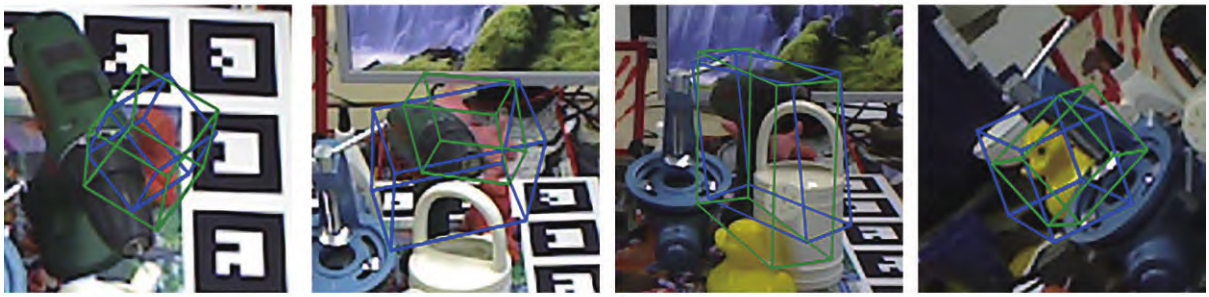
本文在 Occlusion LINEMOD 数据集上进行了遮挡目标的姿态估计测试。在目标被部分遮挡的情况下，本文算法可以正确估计目标的姿态，如图 8(a)～(d)所示。但是当目标存在大面积遮挡或者或在极端角度时，过少的图像特征导致估计失败，如图 8(e)～(h)所示。

表 3 展示了本文算法遮挡目标姿态估计 ADD(-S)测试结果与近期其他算法的对比。其中 Oberweger 等[18]使用热图作为关键点回归的中间形式，并且利用图像分块提升了算法对遮挡的鲁棒性。其余 4 个算法与单目标姿态估计对比算法相同。glue 使用 ADD(-S)度量，其余使用 ADD 度量。加粗数字为本类别中准确率最高的结果。

由表 3 可以看出，与单目标姿态估计结果类似，本文算法在无需后优化算法中有着最高的平均准确率。Oberweger 等[18]虽然对遮挡目标进行了优化，但是本文算法的平均准确率仍然高出 9.4%。PVNet[9]基于目标分割，对遮挡目标姿态估计有天然的优势，但其平均准确率比本文低了 1.2%，可以看出本文算法的性能优越性。但是相比于后处理算法 DeepIM[17]，本文算法在准确率上有所不足。但是 DeepIM 在使用中要先利用 PoseCNN 算法获得姿态初始估计，再进行迭代后优化，复杂的处理流程导致算法无法达到实时处理速度。相比之下，本文算法可以直接通过 P*n*P 算法计算得到姿态，更加简洁高效。PoseCNN 在使用后处理算法优化后，平均准确



(a) Ape正确估计结果示例 (a) Correct example of ape　(b) Cat正确估计结果示例 (b) Correct example of cat　(c) Driller正确估计结果示例 (c) Correct example of driller　(d) Duck正确估计结果示例 (d) Correct example of duck



(e) Ape错误估计结果示例 (e) Incorrect example of ape　(f) Cat错误估计结果示例 (f) Incorrect example of cat　(g) Driller错误估计结果示例 (g) Incorrect example of driller　(h) Duck错误估计结果示例 (h) Incorrect example of duck

图8　部分遮挡目标姿态估计结果(绿色框为真实姿态的目标三维边界框,蓝色框为估计姿态的目标三维边界框)

Fig. 8　Qualitative results for occluded object pose estimation(green 3D bounding boxes represent the ground truth poses, and blue 3D bounding boxes represent our predictions)

表3　遮挡目标姿态估计ADD(-S)指标测试结果

Tab. 3　Occluded object pose estimation ADD(-S) metric test results

算法	Ours	[18]	[6]	[9]	[16]	[16-17]
ape	**23.6**	17.6	2.5	15.8	9.9	**59.2**
can	55.6	53.9	17.5	**63.3**	45.5	**63.5**
cat	16.1	3.3	0.7	**16.7**	0.8	**26.2**
driller	59.5	19.2	1.1	**65.7**	41.6	55.6
duck	**33.7**	62.4	7.7	25.2	19.5	**52.4**
glue*	48.3	39.6	10.1	**49.6**	46.2	**71.7**
hole	**47.3**	21.3	5.5	39.7	27.0	**52.5**
平均值	**40.6**	31.0	6.4	39.4	27.2	**54.4**

*对称对象

率可以提升27.2%,可以说本文算法拥有巨大的准确率提升潜力。

4.6　运行时间实验

本文算法基于目标检测,因此整体算法运行时间与目标检测网络有关。对于640×480大小的图像,使用Inter I7-9700K CPU与NVIDIA RTX3080 GPU时,用yolo-v5进行目标检测的运行时间约为11 ms。本文算法的热图回归网络前向推理用时约为22 ms,热图推理关键点位置约为5 ms,使用EP*n*P算法求解姿态用时约为0.3 ms。算法整体用时约为40 ms,最快运行速率约为25 fps,可以用于实时处理。

4.7　多目标姿态估计问题

本文算法同样可以处理多目标姿态估计问题,可以使用自上而下的姿态估计方法,将多目标姿态估计问题转化为多目标检测问题与多个单目标姿态估计。即首先通过yolo-v5网络同时检测图像中的不同目标,获取不同目标的类别标

签及目标框。将不同目标分别裁剪后，输入相应类别的姿态估计网络中进行姿态估计。而得益于本文算法将目标检测与姿态估计分离的设计，本文算法在进行多目标姿态估计时可以具有与单目标姿态估计相同的准确性，不会使准确性下降。

4.8 损失函数对比实验

为验证本文提出的HWing Loss的有效性，本文进行了不同损失函数的对比实验。使用MSE Loss代替HWing Loss训练热图回归网络，训练参数与4.3节中HWing Loss训练参数相同，同样训练30个轮次，测试结果如表4所示。

表4 对比实验ADD(-S)指标测试对比结果

Tab. 4 Comparison results of comparative experiment ADD (-S) metric test

损失函数	HWing Loss	MSE Loss	HWing Loss	MSE Loss
训练轮次	30	30	10	10
ape	**64.8**	47.3	56.2	41.2
bench	**99.2**	92.4	98.4	47.7
cam	**88.3**	85.2	84.5	64.4
can	**96.8**	93.5	96.9	83.9
cat	**82.7**	69.4	79.5	62.1
driller	**98.0**	91.9	95.1	33.6
duck	**60.1**	44.5	53.8	88.6
eggbox*	**98.9**	98.4	97.7	66.9
glue*	**99.1**	92.8	91.4	86.9
hole	**78.4**	73.6	71.9	57.2
iron	**96.8**	95.8	96.0	93.1
lamp	**97.3**	90.7	92.6	86.9
phone	**93.6**	87.1	92.5	80.4
平均值	**88.8**	81.7	85.1	68.7

*对称对象

可以看出在ADD(-S)指标中，使用HWing Loss相比于使用MSE Loss，平均准确率提升7.1%，每个类别都有所提升。作为对比，表4同时给出了网络分别使用HWing Loss与MSE Loss训练第10轮的实验结果。可以看出，使用HWing Loss训练的网络在训练10轮后的平均准确率已经超越了使用MSE Loss训练30轮的网络。

究其原因是HWing Loss能在训练热图回归网络时，能更关注热图前景像素的误差，并且相比于MSE Loss，HWing Loss在小误差部分有更高的梯度，可以使网络更加快速收敛。

表5给出了使用不同损失函数训练网络预测得到的热图与预期热图的平均像素误差。可以看出，使用HWing Loss训练网络预测热图虽然全局平均误差更大，但是在数量更少且更加重要的前景像素上误差更小。而且误差下降速度更快，10轮训练的结果已经优于使用MSE Loss训练30轮结果，与表4中平均准确率结果一致。由此可以看出，本文提出的HWing Loss相比于MSE Loss更加适合回归热图的训练。

表5 不同损失函数对预测热图误差的影响

Tab. 5 Influence of different loss function on the error of predicting HeatMap

损失函数	HWing Loss		MSE Loss	
训练轮次	30	10	30	10
前景像素误差	0.139 9	0.151 5	0.161 6	0.207 7
全局像素误差	0.007 4	0.009 1	0.001 1	0.001 9

5 结 论

本文分析了用于关键点回归热图的性质，并指出MSE Loss不利于热图回归的问题。为了解决这个问题，本文提出适用于热图回归的Heatmap Wing Loss。同时利用热图性质，改进热图推断关键点的方法。以此为基础，改进基于关键点定位的单目标姿态估计的算法，经实验验证，本文的单目标姿态估计算法在LINEMOD数据集上的ADD(-S)指标平均准确率达到了88.8%，相比于近期其他算法有更好的姿态估计准确率。算法运行速率最快可达到25 fps，适用于实时处理。

参 考 文 献：

[1] RUBLEE E, RABAUD V, KONOLIGE K, *et al*. ORB: an efficient alternative to SIFT or SURF [C]//2011 *International Conference on Computer Vision*. Barcelona: IEEE, 2011: 2564-2571.

[2] BAY H, ESS A, TUYTELAARS T, *et al*. Speeded-up robust features (SURF) [J]. *Computer Vision and Image*

Understanding, 2008, 110(3): 346-359.

[3] 丁南南,刘艳滢,朱明. 尺度相互作用墨西哥帽小波提取图像特征点[J]. 液晶与显示,2012,27(1):125-129.
DING N N, LIU Y Y, ZHU M. Extracting image feature points using scale-interaction of mexican-hat wavelets [J]. *Chinese Journal of Liquid Crystals and Displays*, 2012, 27(1): 125-129. (in Chinese)

[4] DO T T, CAI M, PHAM T, *et al*. Deep-6DPose: recovering 6D object pose from a single RGB image [EB/OL]. (2018-02-28). https://arxiv.org/abs/1802.10367v1.

[5] SUNDERMEYER M, MARTON Z C, DURNER M, *et al*. Augmented autoencoders: implicit 3D orientation learning for 6D object detection [J]. *International Journal of Computer Vision*, 2020, 128(3): 714-729.

[6] TEKIN B, SINHA S N, FUA P, *et al*. Real-time seamless single shot 6D object pose prediction [C]//2018 *IEEE/CVF Conference on Computer Vision and Pattern Recognition*. Salt Lake City: IEEE, 2018: 292-301.

[7] ZHAO Z L, PENG G, WANG H Y, *et al*. Estimating 6D pose from localizing designated surface keypoints [EB/OL]. (2018-12-04). https://arxiv.org/abs/1812.01387.

[8] PARK K, PATTEN T, VINCZE M. Pix2Pose: pixel-wise coordinate regression of objects for 6D pose estimation [C]//2019 *IEEE/CVF International Conference on Computer Vision*. Seoul: IEEE, 2019: 7667-7676.

[9] PENG S D, LIU Y, HUANG Q X, *et al*. PVNet: pixel-wise voting network for 6DoF pose estimation [C]//2019 *IEEE/CVF Conference on Computer Vision and Pattern Recognition*. Long Beach: IEEE, 2019: 4556-4565.

[10] WANG X Y, BO L F, LI F X. Adaptive wing loss for robust face alignment *via* heatmap regression [C]//2019 *IEEE/CVF International Conference on Computer Vision*. Seoul: IEEE, 2019: 6970-6980.

[11] SUN K, XIAO B, LIU D, *et al*. Deep high-resolution representation learning for human pose estimation [C]// *IEEE Conference on Computer Vision and Pattern Recognition*. Long Beach: IEEE, 2019: 5693-5703.

[12] WANG J D, SUN K, CHENG T H, *et al*. Deep high-resolution representation learning for visual recognition [J]. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021, 43(10): 3349-3364.

[13] ZHANG F, ZHU X T, DAI H B, *et al*. Distribution-aware coordinate representation for human pose estimation [C]//2020 *IEEE/CVF Conference on Computer Vision and Pattern Recognition*. Seattle: IEEE, 2020: 7091-7100.

[14] LEPETIT V, MORENO-NOGUER F, FUA P. EP*n*P: an accurate $O(n)$ solution to the P*n*P problem [J]. *International Journal of Computer Vision*, 2009, 81(2): 155-166.

[15] XIAO J X, HAYS J, EHINGER K A, *et al*. SUN database: large-scale scene recognition from abbey to zoo [C]// 2010 *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. San Francisco: IEEE, 2010: 3485-3492.

[16] XIANG Y, SCHMIDT T, NARAYANAN V, *et al*. PoseCNN: a convolutional neural network for 6D object pose estimation in cluttered scenes [C]//*Proceedings of the* 14*th Robotics: Science and Systems*. Pittsburgh:IEEE, 2018.

[17] LI Y, WANG G, JI X Y, *et al*. DeepIM: deep iterative matching for 6D pose estimation [J]. *International Journal of Computer Vision*, 2020, 128(3): 657-678.

[18] OBERWEGER M, RAD M, LEPETIT V. Making deep heatmaps robust to partial occlusions for 3D object pose estimation [C]//*Proceedings of the* 15*th European Conference on Computer Vision*. Munich: Springer, 2018: 125-141.

作者简介:

林　林(1997—),男,内蒙古赤峰人,硕士研究生,2019年于中国科学技术大学获得学士学位,主要从事计算机视觉方面的研究。E-mail:linlin19@mails.ucas.ac.cn

王延杰(1963—),男,吉林长春人,硕士,研究员,1999年于中国科学院长春光学精密机械与物理研究所获得硕士学位,主要从事数字图像处理方面的研究。E-mails:wangyj@ciomp.ac.cn