zero shot pose estimation + cascade predict the 6D Pose (with CAD)

出自

題目

GigaPose: Fast and Robust Novel Object Pose Estimation via One Correspondence

Novel Object Pose Estimation : zero shot，要可以辨別 train 時沒看過的物體

correspondence : 要預測 6d pose，需要找到 2d image 與 3d object model 之間的 correspondence

one correspondence : 一般來說越多點對應 6d pose 越準，但本實驗強調只需 one correspondence 就可以找到 6d pose

Introduction

背景

旋轉軸 :

1. 方位角 (Azimuth) : 繞著 z 軸旋轉 yaw
2. 仰角 (elevation) : 繞著 x 軸旋轉 pitch
3. in-plane rotation : 繞著 y 軸旋轉 roll

動機

1. 不想要每個 object 都重 train
2. 覺得 megapose 太慢了
3. megapose 會遇到 occlusions 容易造成 segmentation 錯誤 的問題

目的

1. cad 從 3d reconstruction 來的，而不是專門請繪師做的
2. 製做 template 時是在 3d space ， 但是只有 yaw 與 pitch 旋轉
3. 用 knn 找出與 input 最近的 template
4. 用 template 來求出 yaw 與 pitch，再用 patch correspondences 算剩下的自由度
5. 利用 template 與 query 的 local feature 求得 x, y
6. 利用 matched point 求得 roll 與 z

結合 templates matching 與 patch matching

將 template 與 image 變成 local feature ⇒ 經過 CNN 但是不 pooling

會形成 W * H * C 的 feature map ，每一個 pixel 都是一個 local feature

因為要對抗 scaling and in-plane rotations

one corresponse ⇒ 因為 Giga 先解掉了兩個自由度，所以剩下四個只要用一個 corresponse 就可以解決了

相關研究

為了達到 zero shot object pose estimation，有兩種方法

1. feature-matching methods : 抽 2D 圖片的 local features 與 given 3D model 對應，再用 pnp 算出 6d pose
2. template matching : 渲染出 3d model 的不同角度與光照行成 template，接著 train model 來找出最近的 template

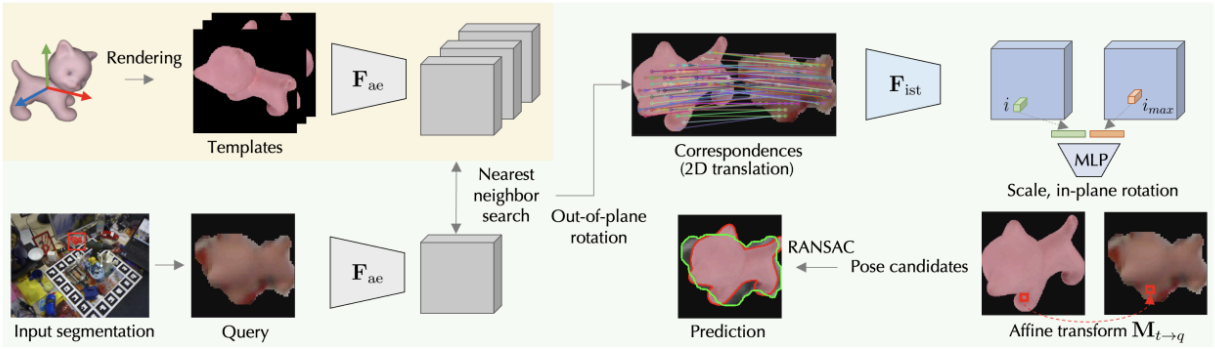
ZS6D 使用 DINOv2 features (FB 的 visual transformer (VIT))

MegaPose : 需要 O(N) 去幫每一組 (query, template) 做 coarse pose estimation

6d pose 預測三步驟 :

1. Object Detection + Segmentation ⇒ CNOS
2. coarse pose estimation ⇒ 缺點 1. 要花很長時間; 缺點 2. 容易被 segmentation 時的錯誤引響而不准
3. refinement ⇒ render-and-compare method

方法



Fae 抽每一個 template 的向量 (一個 object 有 162 個 template)

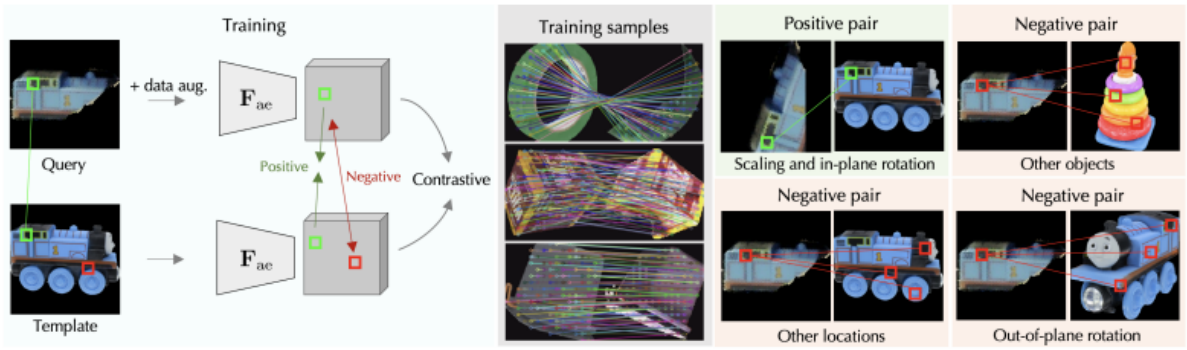
162 是取自於，將正二十面體 (regular icosphere) 上每個三角形的邊取中點往外推到球面後形成的 162 面體 (看起來會像一個球體)，將 object model 放在 球心，在這 162 個頂點往中間拍而成的

query image

1. 先用 CNOS segmentation
2. cropping, padding 接著 resizing

找出 template 與 query 的對應點，將它們放入 Fist 後經過 MLP 就可以算出 in-plane rotation 與 z位移

最後利用 query 的位置求出 x, y ， 6 dof 就完成了



Fae 是 Vision-Transformer (ViT)

local contrastive learning : 越接近的兩個 feature，他們的 output 分數越高

Fae 應該要對 out of plane rotation 敏感, 對 in-plane rotation、 scale、 2D translation 改變時 feature 仍然保持相同

正樣本 : query patch 與 template patch 是同一個物體, 同一組 out of plane 但 different in-plane rotation, scale, and 2D translation

負樣本 : query patch 與 template patch 是不同物體 或是 query patch 與 template patch 是不同組 out of plane 或是 query patch 與 template patch 無法 match (如上面的 other location : 一個在輪胎, 一個在窗戶)

color augmentation along with random cropping and inplane rotation to the input pairs

segmentation mask mQk and mTk ⇒ 遮住 feature map 後就只會剩前景了

InfoNCE loss

$$\mathcal{L}_{\text{out}} = -\sum_{k=1}^{B}\sum_{i=1}^{|\mathbf{m}_{\mathcal{Q}_k}|} \ln \frac{e^{S(\mathbf{q}_k^i, \mathbf{t}_k^{i^*})/\tau}}{\sum_{(k',i')\neq(k,i^*)} e^{S(\mathbf{q}_k^i, \mathbf{t}_{k'}^{i'})/\tau}}$$

S 是 cosine similarity

如何選 template 呢?

1. q (query 經過 Fae 生成的 feature) 上的每一點 i 去找 t (template 經過 Fae) 上最近的一點 i*，
   相近的定義是取自於 Similarity

$$i_{\max} = \arg\max S\left(\mathbf{q}^i, \mathbf{t}^j\right)$$

1. 接著濾除 similarity < 0.5 的 (i, i*) pair 然後把剩下的平均，挑前 K 大的
2. 如果 K != 1，就用 Ransac

Experiments

可以學習的地方

問題