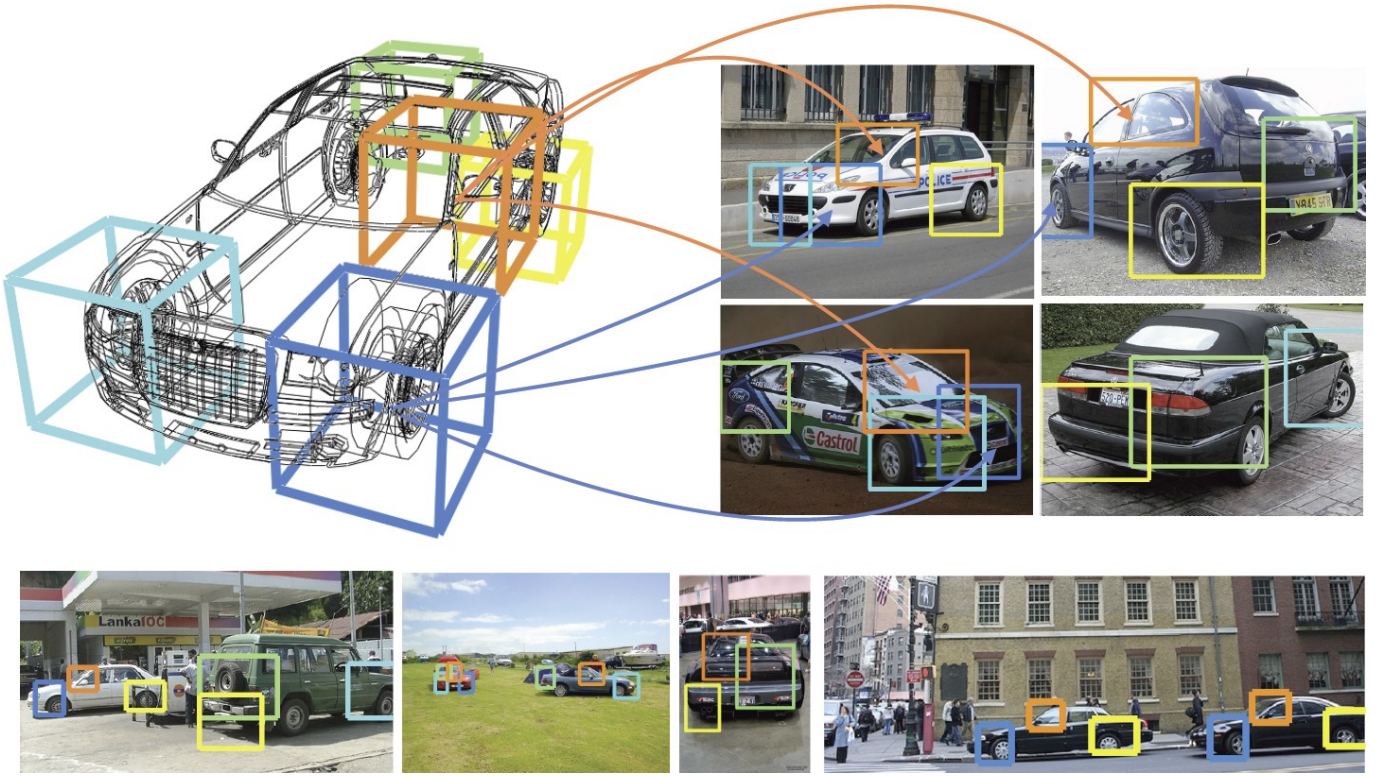
1. Teaching 3D Geometry to Deformable Part Models

CVPR 2012

2. 3D^2PM – 3D Deformable Part Models

ECCV 2012

By Pepik, Gehler, Stark, Schiele at Max Planck Institute
Interpreted by Magnus Burenius at KTH



Background

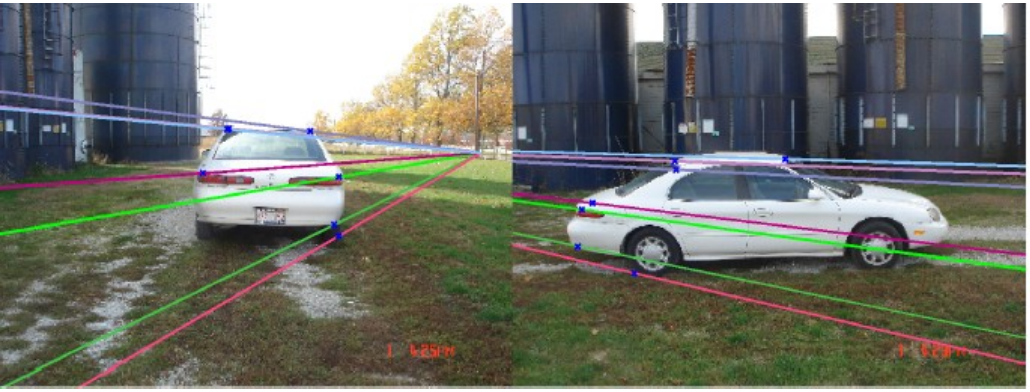
- 3D geometric object class representations have been considered the holy grail of computer vision since its early days.
- Having 3D information improves scene understanding in general.
- While shape-based 3D representations excel in specific domains such as facial pose estimation or marker-less motion capture, they have been largely neglected in favor of less descriptive but more robust 2D local feature-based representations for general object class recognition.
- They want to bridge this gap by extending the most successful 2D bounding box detector, the deformable part model, to 3D.

Concrete Problems

- Detect 2D position of objects.
- Estimate “3D” pose / view-point.
- Wide baseline matching,

  i.e. recover camera motion between two views,

  with relative rotation up to 180°.



Teaching 3D Geometry to Deformable Part Models

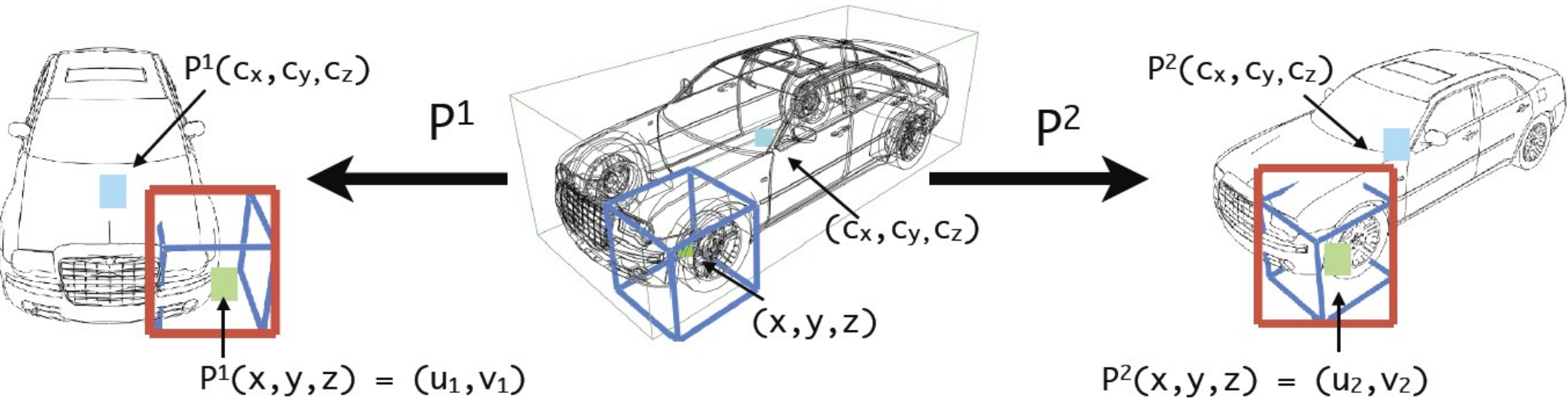
- A. Train independent 2D model for each view.
- B. Train 2D model for each view but use 3D parametrization during training to ensure consistency between parts across views.

3D Deformable Part Model

- C. Use 3D parametrization for both training and testing.

Common for all alternatives (A,B,C)

- They use latent structured SVM for training.
- They test on cars and bicycles.
- They test how 3D CAD models can be used as additional training data. They use 40 models for each class. They use a non-photo-realistic gradient based renderer to get appearances from these for arbitrary views.



A. Train independent 2D model for each view

- Discretize viewpoint into K=24 bins.
- They use the “3D Object classes” data set which has annotations with 8 azimuthal angles and 3 elevations.

  K = 3*8 = 24.
- They thus explicitly handle 2/3 degrees of freedom for rotations in 3D, using spherical coordinates as parametrization.
- Comparison: K=16^3=4096.

A. Train independent 2D model for each view

- Let each viewpoint correspond to a mixture component, which is known during training.
- At test-time the estimated mixture component correspond to the view-point estimate.
- They add a term to the SVM that penalizes wrong view-point estimates, in addition to bounding box location.
- Train similar to ordinary 2D DPM.

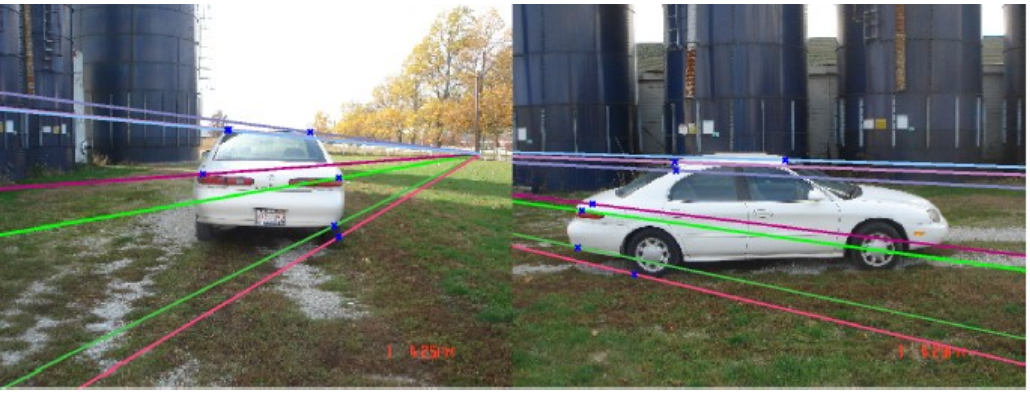
Drawback of A

- The trained parts will not correspond across models / views.

- Why is that important?

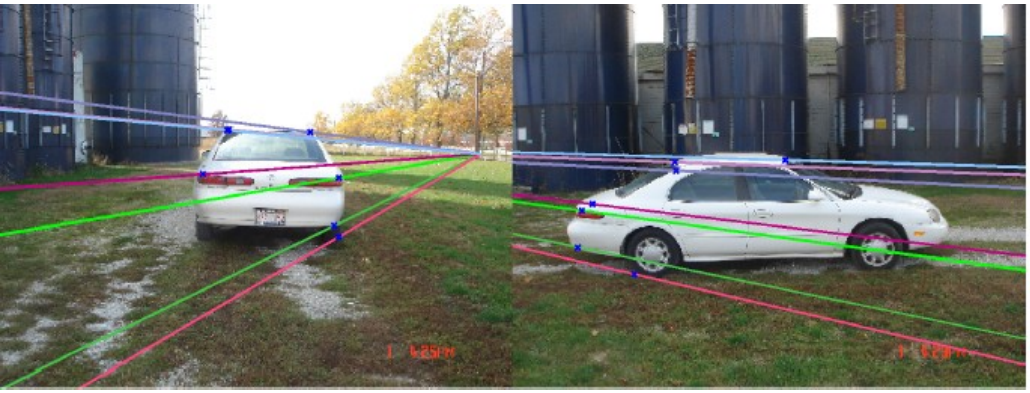
Drawback of A

- Scene understanding in general.
- The estimated part positions cannot be used for computing relative camera motion.
- If we would have corresponding part positions they could be used to estimate the fundamental matrix.



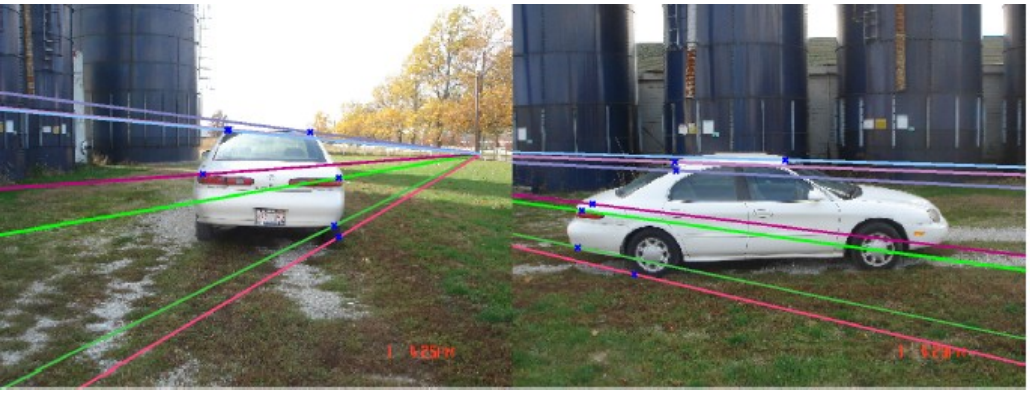
Drawback of A

- But wait. Even if the parts do not correspond we still have the estimate of the 3D rotation of the root and its 2D position in each image.
- Could that be used to compute the scaled orthographic relation between two cameras?



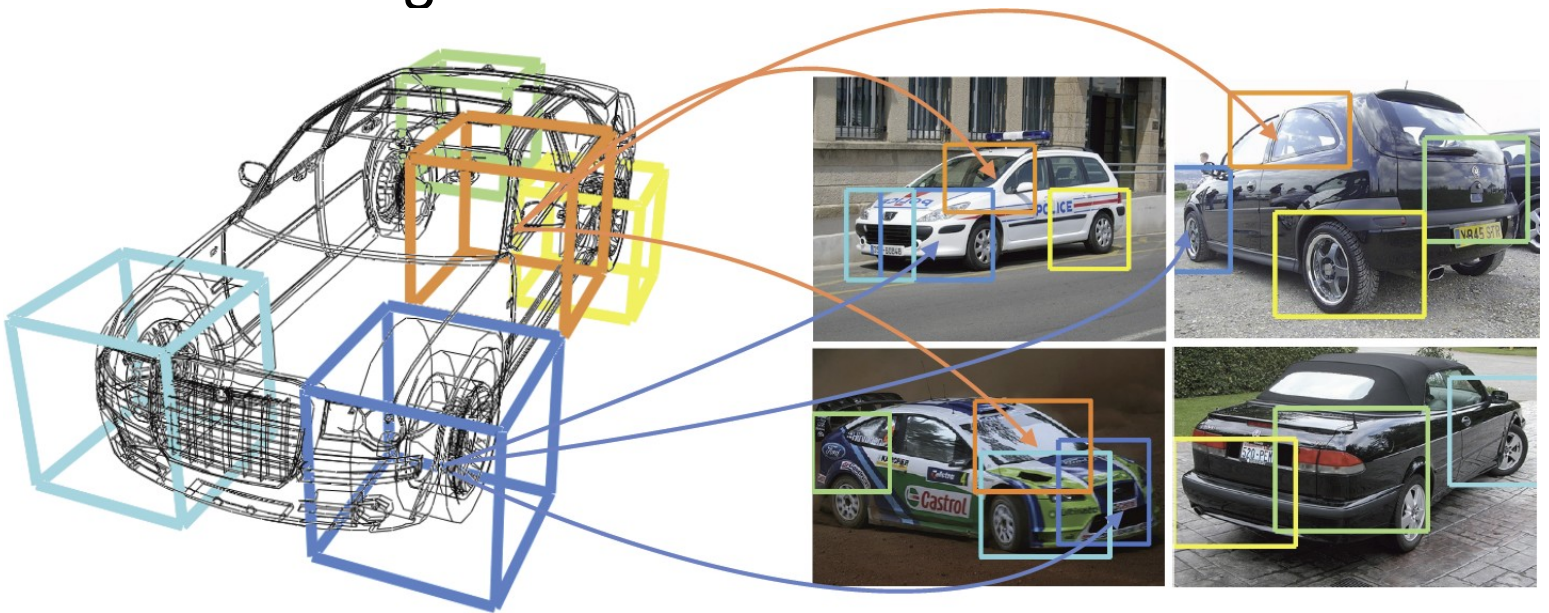
Drawback of A

- Yes, but that does not define 3D translation only 2D translation of camera!
- If we would have corresponding parts, which are free to deform relative their anchors, we could take perspective effects into account.

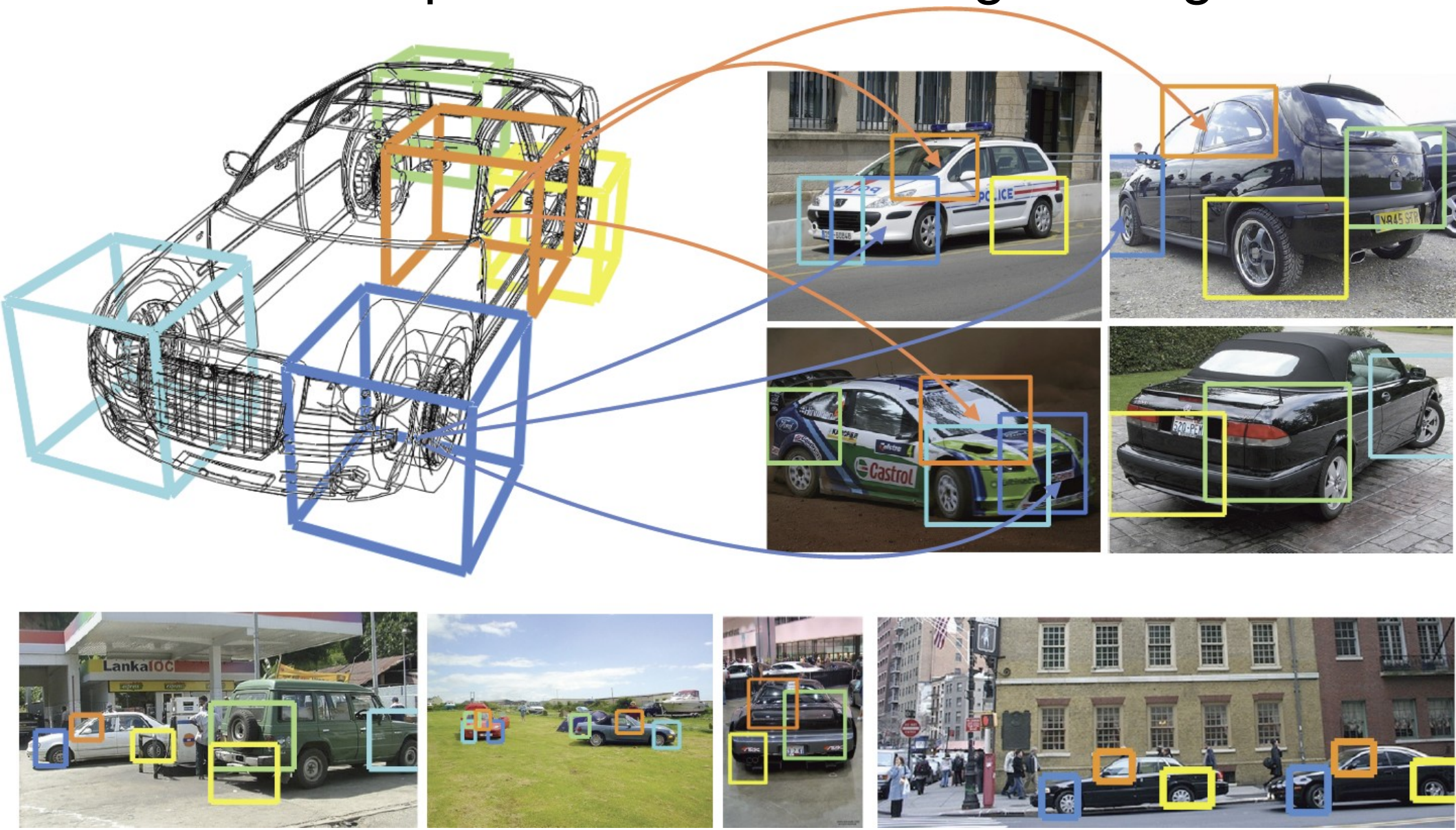


B. 3D parametrization during training

- Each part is represented as a 3D bounding cube.
- The 3D anchor positions of the parts are consistent across views during training.
- For each view / component they learn the 2D deformations relative the image projection of the anchors.
- The result is a 2D model for each view, but the parts now have the same meaning for all of them.



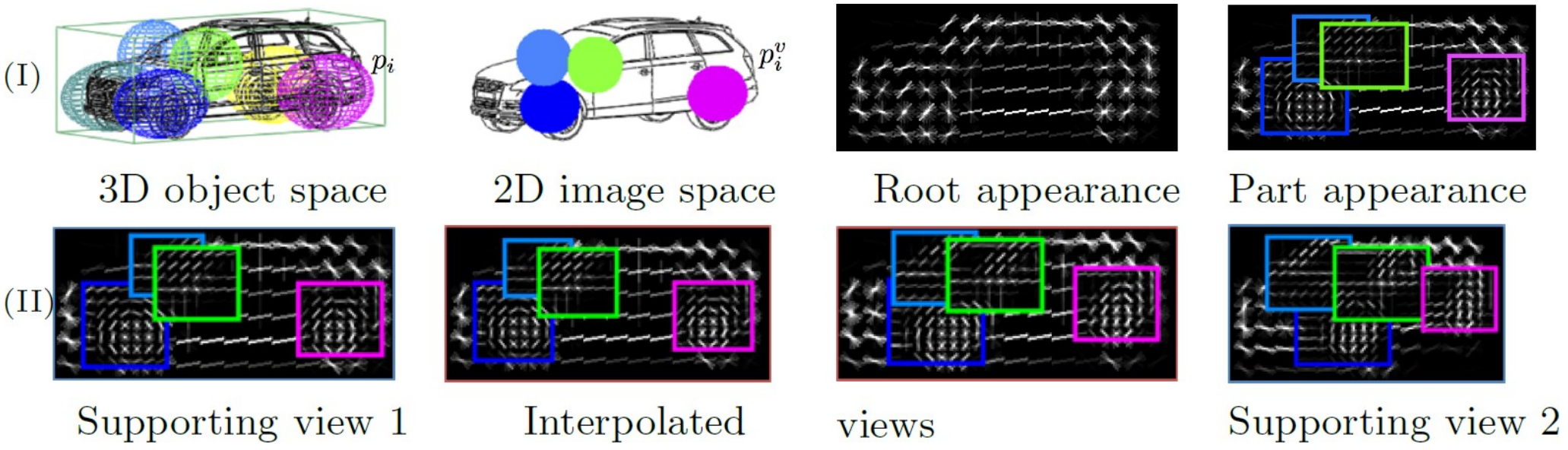
B. 3D parametrization during training



Some results. They estimate the 2D bounding box of parts consistent across views.

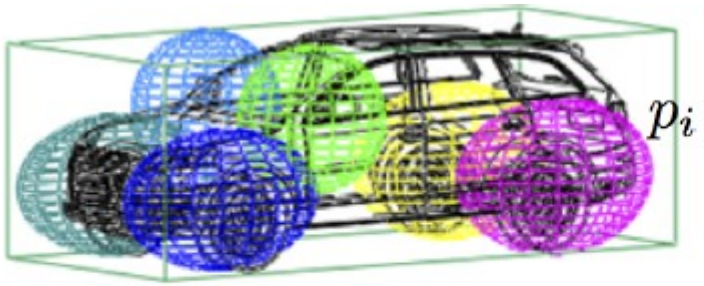
C. 3D Deformable Part Model

- Unlike the previous paper they now consider rotation in a circle, i.e. 1/3 of the degrees of freedom for rotations in 3D. They discretize it in up to K=36 points.
- They learn an appearance model for each part and view-point.
- They interpolate between these to get the appearance model for a part at any continuous angle.

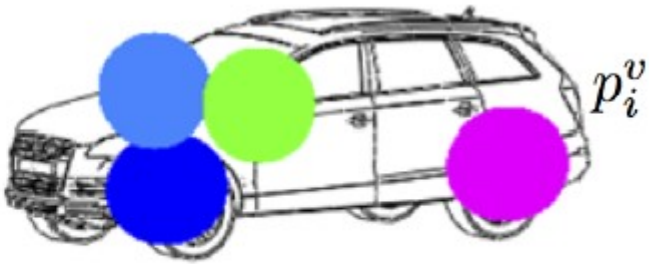


C. 3D Deformable Part Model

- As before they have the anchor of each part in 3D, but now they also learn the deformation of the parts in 3D. This single model is the same for all views.
- They now have 6 deformation parameters per part. Previously in A and B they had 4K deformation parameters per part.
- At test-time they can project these anchors and deformations to any hypothetical camera angle. Thus, unlike the two previous approaches they are not restricted to use the view-discretization of the training when testing.



3D object space



2D image space

C. 3D Deformable Part Model

Some results. Note that they now estimate 3D bounding cube for each part.

Quantitative Results

- For quantitative results have a look at the papers.

Future work

- How can this be generalized to full 3D rotations?
- Would it scale well?

Future work

- How can this be generalized to articulated objects, like humans, described by tree graphs instead of star graphs, where each part can rotate in 3D.
- Would it scale well?