Contents lists available at ScienceDirect

Computer Vision and Image Understanding

journal homepage: www.elsevier.com/locate/cviu

Learning to generate novel views of objects for class recognition

Han-Pang Chiu *, Leslie Pack Kaelbling, Tomás Lozano-Pérez

Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge, MA 02139, USA

ARTICLE INFO

Article history:
Received 14 July 2008
Accepted 10 June 2009
Available online 12 August 2009

Keywords:
Object recognition
Categorization
Shape representation

ABSTRACT

Multi-view object class recognition can be achieved using existing approaches for single-view object class recognition, by treating different views as entirely independent classes. This strategy requires a large amount of training data for many viewpoints, which can be costly to obtain. We describe a method for constructing a weak three-dimensional model from as few as two views of an object of the target class, and using that model to transform images of objects from one view to several other views, effectively multiplying their value for class recognition. Our approach can be coupled with any 2D image-based recognition system. We show that automatically transformed images dramatically decrease the data requirements for multi-view object class recognition.

1. Introduction

After seeing a number of instances of objects from the same class from different views, a vision system should be able to recognize other instances of that class from arbitrary views, including views that have not been seen before. In most current approaches to object class recognition, the problem of recognizing multiple views of the same object class is treated as one of recognizing multiple independent object classes, with a separate model learned for each. This independent-view approach can be effective when there are many instances at different views available for training, but can typically only handle new viewpoints that are a small distance from some view on which it has been trained.

An alternate strategy is to use multiple views of multiple instances to construct a model of the distribution of three-dimensional shapes of the object class. Such a model would allow recognition of many entirely novel views. This is a very difficult problem, which is as yet unsolved.

In our work, we take an intermediate approach, which exploits the fact that there is a fundamental relationship between multiple views of the same class of objects, and allows subsequent unlabeled views of new objects from the same class to be transformed into novel views. We define the *Potemkin*[1] *model* of a three-dimensional object as a collection of parts, which are oriented 3D primitive shapes. The model allows the parts to have an arbitrary arrangement in 3D, but assumes that, from any viewpoint, the parts themselves can be treated as being nearly planar. We will refer to the arrangement of the part centroids in 3D as the *skeleton* of the object. As one moves the viewpoint around the object, the part centroids move as dictated by the 3D skeleton; but rather than having the detailed 3D shape model that would be necessary for predicting each part's shape in each view, we model the change of each part's image between views as a 2D perspective transformation.

The Potemkin model is trained and used in three phases. The first phase is class-independent and carried out once only. In it, the system learns, for each element of a set of simple oriented 3D primitive shapes, what 2D image transformations are induced by changes of viewpoint of the shape primitive. The necessary data can be relatively simply acquired from synthetic image sequences of a few objects rotating through the desired space of views. After the first phase is complete, the model can be used for new object classes with very little overhead. This process might be seen as learning basic view transformations of primitive 3D shapes by "watching the world go by", and then using that knowledge to accelerate learning about new object types which are constituted of these primitives.

The second phase is class-specific: a few images from a target class (typically of a single object from two views) are used to estimate the 3D skeleton of the object class, to select an oriented primitive 3D shape for each of the parts, and to initialize image transforms between pairs of views, for each part.

The third phase is view-specific: given a new view of interest, the class skeleton is projected into that view, specifying the 2D centroids of the parts. Then, all available 2D training images, from any view, are transformed into this view using the image transforms selected in the second phase. These new images are "virtual training examples" of previously seen objects from novel views. These virtual training examples, along with any real training examples that are available, can then be used as training data for any view-dependent 2D recognition system, which can then be used

* Corresponding author.
E-mail address: chiu@csail.mit.edu (H.-P. Chiu).

[1] So-called 'Potemkin villages' were artificial villages, constructed only of facades. Our models, too, are constructed of facades.

1077-3142/$ - see front matter © 2009 Elsevier Inc. All rights reserved.
doi:10.1016/j.cviu.2009.06.004

to detect instances of the object class in the novel viewpoint. In our experiments, we have trained a recognizer for a novel view of an object with 100 virtual images transformed from 100 training examples in other views; that recognizer, even though it had never seen an actual image of an object of this class from this view, performs as well as the same recognition algorithm trained on 50 actual images from this view.

In the following section we provide an overview of related work. Then in Section 3 we sketch a basic version of the Potemkin model, which uses only a single oriented primitive for learning the transforms from view to view in the first phase. These transforms are then used as the basis for transforms of all parts in the model. This basic model is ultimately too weak to describe many object classes well. Section 4 extends the basic Potemkin model to use a basis set of multiple oriented primitives in the first phase, and to select among them to represent each of the parts of the target class based on a pair of initial training images of the class. For each part, the transforms learned from the selected primitive are used to initialize image transforms between pairs of views. In Section 5 we provide a self-supervised labeling method that obviates the need for most of part labeling in the real training data. Section 6 demonstrates that these extensions significantly outperform the basic model and enable existing view-dependent detection systems to achieve the same level of performance with many fewer real training images required.

2. Related work

There has been a great deal of work on modeling and recognition for single 3D objects (e.g., [1–4]), but these methods are not designed to cover the variability in shape and appearance among instances of a class and cannot be readily generalized for object class recognition. There is also a long tradition of representing objects as arrangements of 3D building blocks (e.g., [5–11]). However, it has been difficult to achieve robust recognition performance in real images using these approaches.

Most recent advances in object class recognition attempt to detect examples of a target class from only a single viewpoint [12–22]. Our work leverages such advances, and enables them to be applied to multi-view recognition without excessive training-data requirements.

There is an existing body of work on multi-view object class recognition for specific classes such as cars and faces [23–28]. For example, Everingham and Zisserman [29] generate virtual training data for face recognition from multiple viewpoints using a reconstructed 3D model of the head.

For object recognition, more generally, there are a number of effective multi-view object class recognition systems that are constructed from several independent single-view detectors. Crandall and Huttenlocher [30] treat ranges of poses as separate classes, with a separate model learned for each. Torralba et al. [31] show that sharing features among models for multiple views of the same object class improves both running time and recognition performance. Leibe et al.'s detection system [32] for cars combines a set of seven view-dependent ISM detectors [33], each of which requires hundreds of training examples to be effective.

There is increased recent interest in methods that make deeper use of the relationship between views of the same class, via a range of approaches.

One approach is to link regions of the same or similar training instances across different viewpoints, forming a multi-view class model for recognition. Thomas et al. [34] construct separate view-dependent detectors for each of the viewing directions, and track regions of the same instance across different viewpoints. They use these integrated view-dependent detectors to improve detection, by transferring information from one viewpoint to another. Savarese and Fei-Fei [35] also relate multiple views by finding correspondences among 2D regions, formed by feature grouping, on the same instance across different views. Compared to the work of Thomas et al, these models are more compact and can be learned with very weak supervision. Rather than constructing multiple view-dependent detectors, Kushal et al. [36] construct a single model that relates partial surfaces of the target object class among multiple viewpoints. Each partial surface is formed by locally dense rigid assemblies of image features on instances across different views. Each of these three methods requires many training images from each viewpoint and the first two methods require many views of the same training instances, which are relatively difficult to obtain.

An alternative approach is to construct explicit 3D shape models of object classes. Hoeim et al. [37] create a coarse 3D model from mean segmentations of two views of training instances, and avoid the need for multiple views of the same training object instance during learning. Yan et al. [38] use training images, taken from a dense sampling of viewpoints around a specific object, to reconstruct a 3D model for the target class. Both methods form correspondences between 2D features across training instances at arbitrary viewpoints by projecting the surfaces of the 3D model onto each instance. Thus, the features can be shared across viewpoints through 3D correspondence. However, these systems still require many training instances at many viewpoints to achieve satisfactory recognition results. To avoid collecting real training images, Liebelt et al. [39] use a database of 3D CAD models of objects of the target class, and construct feature descriptors on synthetic images generated from these 3D models. However, 3D CAD models with realistic textures are not easily obtained for some object classes.

The Potemkin model [40] is intermediate between the approaches based on cross-view constraints and those based on detailed 3D models. Cross-view constraints, which are formed by linking 2D regions of instances from one viewpoint to another, require many training images in each of the modeled viewpoints. Detailed 3D models of variable object classes can be difficult to learn and use. Therefore, we construct a relatively weak 3D model, which can be learned from as few as two labeled images, but is powerful enough to generate virtual examples in novel viewpoints, amplifying the effectiveness of any method that needs images from multiple views.

3. Basic shape model

The Potemkin model is used to represent the approximate 3D shape and relationship among views of a class of objects. In this section we describe the basic version of the Potemkin model in detail, including how to estimate the parameters from a small number of training images, and use it to generate virtual training images in novel views. The basic model is ultimately too weak to describe many classes of objects well, because it has only a single primitive part shape. We will extend it to include multiple primitive shapes and self-occlusion in Section 4. In addition, the training algorithm for the basic model requires the outlines of the parts in all real training images to be labeled. We will describe a simple approach that obviates the need for most of this labeling but still requires a decomposition into parts in Section 5.

3.1. The basic Potemkin model

Informally, a basic Potemkin model can be viewed as a collection of planar ''facades'', one for each part, which are arranged in three dimensions. In order to model the transformation of an