Cascaded Pose Regression

Piotr Dollár Peter Welinder Pietro Perona
California Institute of Technology
{pdollar,welinder,perona}@caltech.edu

Abstract

We present a fast and accurate algorithm for computing the 2D pose of objects in images called cascaded pose regression (CPR). CPR progressively refines a loosely specified initial guess, where each refinement is carried out by a different regressor. Each regressor performs simple image measurements that are dependent on the output of the previous regressors; the entire system is automatically learned from human annotated training examples. CPR is not restricted to rigid transformations: 'pose' is any parameterized variation of the object's appearance such as the degrees of freedom of deformable and articulated objects. We compare CPR against both standard regression techniques and human performance (computed from redundant human annotations). Experiments on three diverse datasets (mice, faces, fish) suggest CPR is fast (2-3ms per pose estimate), accurate (approaching human performance), and easy to train from small amounts of labeled data.

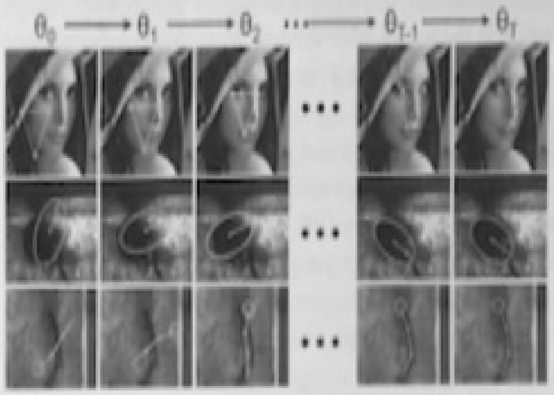

Figure 1. Object pose (green wire frame) is computed by cascaded pose regression (CPR) starting from a coarse initial guess (orange wire frame). The parameterization of pose is arbitrary and need only be consistent across training examples. CPR is implemented as a sequence of regressors progressively refining the estimate of the pose θ. At each step $t = 1 \ldots T$ in the cascade, a regressor R^t computes a new pose estimate θ_t from the image and from the previous regressor's estimate θ_{t-1}. *Left:* Initial guess θ_0; *Right:* final estimate θ_T. Each row shows a test case culled from three different data sets. The same CPR code was trained to compute the pose of different objects/categories from a relatively small sample of hand-annotated training examples.

1. Introduction

Detection and localization are among the most useful functions of vision. Detection consists of giving a one-bit answer to the question "*Is object/category x in the image?*". Localization is a more subtle problem: in its simplest and most popular form [11], it consists of identifying the smallest rectangular region of the image that contains the object in question. This is perfectly sufficient for categories whose main geometric degrees of freedom in the image are translation and scale, such as frontal faces and pedestrians. More generally, one wishes to recover *pose*, that is a number of parameters that influence the image of the object. Most commonly pose refers to geometric transformations of rigid objects [23] including the configuration of articulated objects, for example the limbs of a human body [26, 14] or vehicle layout [21]. More broadly, pose is any set of systematic and parameterizable changes in the appearance of the object [5]. There are two distinct reasons for comput-

pose explicitly, (2) pose is the desired output of the vision module. In this work we are interested in the latter: we wish to estimate the pose of an object given its rough initial location, for example as provided by a tracker.

The predominant approach for object localization in position and scale is to use a 'sliding window', *i.e.*, repeating a binary classification task, "*Is object x at location y?*", for a fine-grained sampling of the pose parameters. Although this generates a large number of tests, sliding window methods can be made more efficient through cascades [28], distance transforms [13], branch-and-bound search [20] and coarse to fine approaches [15]. Such methods can can be extended to more complex notions of pose by repeatedly answering queries of the form "*Is object x at location y with pose θ?*", one for each partition of the pose θ. For example, for face detection it is common to train a separate classifier for different levels of out of plane rotation [28]. Of course this

Cascaded Pose Regression

Piotr Dollár Peter Welinder Pietro Perona
California Institute of Technology
{pdollar,welinder,perona}@caltech.edu

Abstract

We present a fast and accurate algorithm for computing the 2D pose of objects in images called cascaded pose regression (CPR). CPR progressively refines a loosely specified initial guess, where each refinement is carried out by a different regressor. Each regressor performs simple image measurements that are dependent on the output of the previous regressors; the entire system is automatically learned from human annotated training examples. CPR is not restricted to rigid transformations: 'pose' is any parameterized variation of the object's appearance such as the degrees of freedom of deformable and articulated objects. We compare CPR against both standard regression techniques and human performance (computed from redundant human annotations). Experiments on three diverse datasets (mice, faces, fish) suggest CPR is fast (2-3ms per pose estimate), accurate (approaching human performance), and easy to train from small amounts of labeled data.

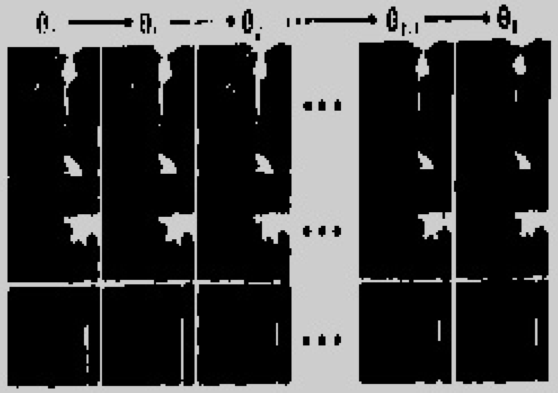

Figure 1. Object pose (green wire frame) is computed by cascaded pose regression (CPR) starting from a coarse initial guess (orange wire frame). The parameterization of pose is arbitrary and need only be consistent across training examples. CPR is implemented as a sequence of regressors progressively refining the estimate of the pose θ. At each step $t = 1 \dots T$ in the cascade, a regressor R^t computes a new pose estimate θ_t from the image and from the previous regressor's estimate θ_{t-1}. *Left:* Initial pose θ_0. *Right:* final estimate θ_T. Each row shows a test case culled from three different datasets. The same CPR code was trained to compute the pose of different objects/categories from a relatively small sample of hand annotated training examples.

1. Introduction

Detection and localization are among the most useful functions of vision. Detection consists of giving a one bit answer to the question "*Is object/category x in the image?*". Localization is a more subtle problem: in its simplest and most popular form [ ], it consists of identifying the smallest rectangular region of the image that contains the object in question. This is perfectly sufficient for categories whose main geometric degrees of freedom in the image are translation and scale, such as frontal faces and pedestrians. More generally, one wishes to recover *pose*, that is a number of parameters that influence the image of the object. Most commonly pose refers to geometric transformations of rigid objects [ ], including the configuration of articulated objects, for example the limbs of a human body [,] or vehicle layout [ ]. More broadly, pose is any set of systematic and parameterizable changes in the appearance of the object [ ]. There are two distinct reasons for computing pose explicitly, (2) pose is the desired output of the vision module. In this work we are interested in the latter: we wish to estimate the pose of an object given its rough initial location, for example as provided by a tracker.

The predominant approach for object localization in position and scale is to use a 'sliding window', i.e., repeating a binary classification task, "*Is object x at location y?*", for a fine-grained sampling of the pose parameters. Although this generates a large number of tests, sliding window methods can be made more efficient through cascades [ ], distance transforms [ ], branch and bound search [ ] and coarse to fine approaches [ ]. Such methods can can be extended to more complex notions of pose by repeatedly answering queries of the form "*Is object x at location y with pose θ?*", one for each partition of the pose θ. For example, for face detection it is common to train a separate classifier for different levels of out of plane rotation [ ]. Of course this