The Conditional Lucas & Kanade Algorithm

Chen-Hsuan Lin, Rui Zhu, and Simon Lucey

The Robotics Institute
Carnegie Mellon University
Pittsburgh, PA 15213
{chenhsul,rz1}@andrew.cmu.edu, slucey@cs.cmu.edu

Abstract. The Lucas & Kanade (LK) algorithm is the method of choice for efficient dense image and object alignment. The approach is efficient as it attempts to model the connection between appearance and geometric displacement through a linear relationship that assumes independence across pixel coordinates. A drawback of the approach, however, is its generative nature. Specifically, its performance is tightly coupled with how well the linear model can synthesize appearance from geometric displacement, even though the alignment task itself is associated with the inverse problem. In this paper, we present a new approach, referred to as the Conditional LK algorithm, which: (i) directly learns linear models that predict geometric displacement as a function of appearance, and (ii) employs a novel strategy for ensuring that the generative pixel independence assumption can still be taken advantage of. We demonstrate that our approach exhibits superior performance to classical generative forms of the LK algorithm. Furthermore, we demonstrate its comparable performance to state-of-the-art methods such as the Supervised Descent Method with substantially less training examples, as well as the unique ability to "swap" geometric warp functions without having to retrain from scratch. Finally, from a theoretical perspective, our approach hints at possible redundancies that exist in current state-of-the-art methods for alignment that could be leveraged in vision systems of the future.

Keywords: Image alignment, Lucas & Kanade, Supervised Descent Method

1 Introduction

The Lucas & Kanade (LK) algorithm [9] has been a popular approach for tackling dense alignment problems for images and objects. At the heart of the algorithm is the assumption that an approximate linear relationship exists between pixel appearance and geometric displacement. Such a relationship is seldom exactly linear, so a linearization process is typically repeated until convergence. Pixel intensities are not deterministically differentiable with respect to geometric displacement; instead, the linear relationship must be established stochastically through a learning process. One of the most notable properties of the LK algorithm is how efficiently this linear relationship can be estimated. This efficiency stems from the assumption of independence across pixel coordinates - the parameters describing this linear relationship are classically referred to as image gradients. In practice, these image gradients are estimated through finite differencing operations. Numerous extensions and variations upon the LK algorithm have

subsequently been explored in literature [3], and recent work has also demonstrated the utility of the LK framework [2, 4, 1] using classical dense descriptors such as dense SIFT [8], HOG [5], and LBP [12].

A drawback to the LK algorithm and its variants, however, is its generative nature. Specifically, it attempts to synthesize, through a linear model, how appearance changes as a function of geometric displacement, even though its end goal is the inverse problem. Recently, Xiong & De la Torre [14, 16, 15] proposed a new approach to image alignment known as the Supervised Descent Method (SDM). SDM shares similar properties with the LK algorithm as it also attempts to establish the relationship between appearance and geometric displacement using a sequence of linear models. One marked difference, however, is that SDM directly learns how geometric displacement changes as a function of appearance. This can be viewed as estimating the conditional likelihood function $p(\mathbf{y}|\mathbf{x})$, where $\mathbf{y}$ and $\mathbf{x}$ are geometric displacement and appearance respectively. As reported in literature [7] (and also confirmed by our own experiments in this paper), this can lead to substantially improved performance over classical LK as the learning algorithm is focused directly on the end goal (*i.e.* estimating geometric displacement from appearance).

Although it exhibits many favorable properties, SDM also comes with disadvantages. Specifically, due to its non-generative nature, SDM cannot take advantage of the pixel independence assumption enjoyed through classical LK (see Section 4 for a full treatment on this asymmetric property). Instead, it needs to model full dependence across all pixels, which requires: (i) a large amount of training data, and (ii) the requirement of adhoc regularization strategies in order to avoid a poorly conditioned linear system. Furthermore, SDM does not utilize prior knowledge of the type of geometric warp function being employed (*e.g.* similarity, affine, homography, point distribution model, etc.), which further simplifies the learning problem in classical LK.

In this paper, we propose a novel approach which, like SDM, attempts to learn a linear relationship between geometric displacement directly as a function of appearance. However, unlike SDM, we enforce that the pseudo-inverse of this linear relationship enjoys the generative independence assumption across pixels while utilizing prior knowledge of the parametric form of the geometric warp. We refer to our proposed approach as the Conditional LK algorithm. Experiments demonstrate that our approach achieves comparable, and in many cases better, performance to SDM across a myriad of tasks with substantially less training examples. We also show that our approach does not require any adhoc regularization term, and it exhibits a unique property of being able to "swap" the type of warp function being modeled (*e.g.* replace a homography with an affine warp function) without the need to retrain. Finally, our approach offers some unique theoretical insights into the redundancies that exist when attempting to learn efficient object/image aligners through a conditional paradigm.

Notations. We define our notations throughout the paper as follows: lowercase boldface symbols (*e.g.* $\mathbf{x}$) denote vectors, uppercase boldface symbols (*e.g.* $\mathbf{R}$) denote matrices, and uppercase calligraphic symbols (*e.g.* $\mathcal{I}$) denote functions. We treat images as a function of the warp parameters, and we use the notations $\mathcal{I}(\mathbf{x}) : \mathbb{R}^2 \to \mathbb{R}^K$ to indicate sampling of the K-channel image representation at subpixel location $\mathbf{x} = [x, y]^\top$. Common examples of multi-channel image representations include descriptors such as

dense SIFT, HOG and LBP. We assume $K = 1$ when dealing with raw grayscale images.

2 The Lucas & Kanade Algorithm

At its heart, the Lucas & Kanade (LK) algorithm utilizes the assumption that,

$$\mathcal{I}(\mathbf{x} + \Delta\mathbf{x}) \approx \mathcal{I}(\mathbf{x}) + \nabla\mathcal{I}(\mathbf{x})\Delta\mathbf{x} \ . \tag{1}$$

where $\mathcal{I}(\mathbf{x}) : \mathbb{R}^2 \rightarrow \mathbb{R}^K$ is the image function representation and $\nabla\mathcal{I}(\mathbf{x}) : \mathbb{R}^2 \rightarrow \mathbb{R}^{K\times 2}$ is the image gradient function at pixel coordinate $\mathbf{x} = [x, y]$. In most instances, a useful image gradient function $\nabla\mathcal{I}(\mathbf{x})$ can be efficiently estimated through finite differencing operations. An alternative strategy is to treat the problem of gradient estimation as a per-pixel linear regression problem, where pixel intensities are samples around a neighborhood in order to "learn" the image gradients [4]. A focus of this paper is to explore this idea further by examining more sophisticated conditional learning objectives for learning image gradients.

For a given geometric warp function $\mathcal{W}\{\mathbf{x}; \mathbf{p}\} : \mathbb{R}^2 \rightarrow \mathbb{R}^2$ parameterized by the warp parameters $\mathbf{p} \in \mathbb{R}^P$, one can thus express the classic LK algorithm as minimizing the sum of squared differences (SSD) objective,

$$\min_{\Delta\mathbf{p}} \sum_{d=1}^{D} \left\| \mathcal{I}(\mathcal{W}\{\mathbf{x}_d; \mathbf{p}\}) + \nabla\mathcal{I}(\mathcal{W}\{\mathbf{x}_d; \mathbf{p}\}) \frac{\partial\mathcal{W}(\mathbf{x}_d; \mathbf{p})}{\partial\mathbf{p}^\top} \Delta\mathbf{p} - \mathcal{T}(\mathbf{x}_d) \right\|_2^2 , \tag{2}$$

which can be viewed as a quasi-Newton update. The parameter $\mathbf{p}$ is the initial warp estimate, $\Delta\mathbf{p}$ is the warp update being estimated, and $\mathcal{T}$ is the template image we desire to align the source image $\mathcal{I}$ against. The pixel coordinates $\{\mathbf{x}_d\}_{d=1}^D$ are taken with respect to the template image's coordinate frame, and $\frac{\partial\mathcal{W}(\mathbf{x};\mathbf{p})}{\partial\mathbf{p}^\top} : \mathbb{R}^2 \rightarrow \mathbb{R}^{2\times P}$ is the warp Jacobian. After solving Equation 2, the current warp estimate has the following additive update,

$$\mathbf{p} \leftarrow \mathbf{p} + \Delta\mathbf{p} \ . \tag{3}$$

As the relationship between appearance and geometric deformation is not solely linear, Equations 2 and 3 must be applied iteratively until convergence is achieved.

Inverse compositional fitting. The canonical LK formulation presented in the previous section is sometimes referred to as the forwards additive (FA) algorithm [3]. A fundamental problem with the forwards additive approach is that it requires recomputing the image gradient and warp Jacobian in each iteration, greatly impacting computational efficiency. Baker and Matthews [3] devised a computationally efficient extension to forwards additive LK, which they refer to as the inverse compositional (IC) algorithm. The IC-LK algorithm attempts to iteratively solve the objective

$$\min_{\Delta\mathbf{p}} \sum_{d=1}^{D} \left\| \mathcal{I}(\mathcal{W}\{\mathbf{x}_d; \mathbf{p}\}) - \mathcal{T}(\mathbf{x}_d) - \nabla\mathcal{T}(\mathbf{x}_d) \frac{\partial\mathcal{W}(\mathbf{x}_d; \mathbf{0})}{\partial\mathbf{p}^\top} \Delta\mathbf{p} \right\|_2^2 , \tag{4}$$

followed by the inverse compositional update

$$\mathbf{p} \leftarrow \mathbf{p} \circ (\Delta\mathbf{p})^{-1}, \tag{5}$$

where we have abbreviated the notation $\circ$ to be the composition of warp functions parametrized by $\mathbf{p}$, and $(\Delta\mathbf{p})^{-1}$ to be the parameters of the inverse warp function parametrized by $\Delta\mathbf{p}$. We can express Equation 4 in vector form as

$$\min_{\Delta\mathbf{p}} \|\mathcal{I}(\mathbf{p}) - \mathcal{T}(\mathbf{0}) - \mathbf{W}\Delta\mathbf{p}\|_2^2, \tag{6}$$

where,

$$\mathbf{W} = \begin{bmatrix} \nabla\mathcal{T}(\mathbf{x}_1) \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots \nabla\mathcal{T}(\mathbf{x}_D) \end{bmatrix} \begin{bmatrix} \frac{\partial\mathcal{W}(\mathbf{x}_1;\mathbf{0})}{\partial\mathbf{p}^\top} \\ \vdots \\ \frac{\partial\mathcal{W}(\mathbf{x}_D;\mathbf{0})}{\partial\mathbf{p}^\top} \end{bmatrix}$$

and

$$\mathcal{I}(\mathbf{p}) = \begin{bmatrix} \mathcal{I}(\mathcal{W}\{\mathbf{x}_1;\mathbf{p}\}) \\ \vdots \\ \mathcal{I}(\mathcal{W}\{\mathbf{x}_D;\mathbf{p}\}) \end{bmatrix}, \quad \mathcal{T}(\mathbf{0}) = \begin{bmatrix} \mathcal{T}(\mathcal{W}\{\mathbf{x}_1;\mathbf{0}\}) \\ \vdots \\ \mathcal{T}(\mathcal{W}\{\mathbf{x}_D;\mathbf{0}\}) \end{bmatrix}.$$

Here, $\mathbf{p} = \mathbf{0}$ is considered the identity warp (*i.e.* $\mathcal{W}\{\mathbf{x};\mathbf{0}\} = \mathbf{x}$). It is easy to show that the solution to Equation 6 is given by

$$\Delta\mathbf{p} = \mathbf{R}[\mathcal{I}(\mathbf{p}) - \mathcal{T}(\mathbf{0})], \tag{7}$$

where $\mathbf{R} = \mathbf{W}^\dagger$. The superscript † denotes the Moore-Penrose pseudo-inverse operator. The IC form of the LK algorithm comes with a great advantage: the gradients $\nabla\mathcal{T}(\mathbf{x})$ and warp Jacobian $\frac{\partial\mathcal{W}(\mathbf{x};\mathbf{0})}{\partial\mathbf{p}^\top}$ are evaluated at the identity warp $\mathbf{p} = \mathbf{0}$, regardless of the iterations and the current state of $\mathbf{p}$. This means that $\mathbf{R}$ remains constant across all iterations, making it advantageous over other variants in terms of computational complexity. For the rest of this paper, we shall focus on the IC form of the LK algorithm.

3 Supervised Descent Method

Despite exhibiting good performance on many image alignment tasks, the LK algorithm can be problematic to use when there is no specific template image $\mathcal{T}$ to align against. For many applications, one may be given just an ensemble of M ground-truth images and warps $\{\mathcal{I}_m, \mathbf{p}_m\}_{m=1}^M$ of the object of interest. If one has prior knowledge of the distribution of warp displacements to be encountered, one can synthetically generate N examples to form a much larger set $\mathcal{S} = \{\Delta\mathbf{p}_n, \mathcal{I}_n(\mathbf{p}_n \circ \Delta\mathbf{p}_n)\}_{n=1}^N$ to learn from, where $N \gg M$. In these circumstances, a strategy recently put forward known as the Supervised Descent Method (SDM) [14] has exhibited state-of-the-art performance across a number of alignment tasks, most notably facial landmark alignment. The approach attempts to directly learn a regression matrix that minimizes the following SSD objective,

$$\min_{\mathbf{R}} \sum_{n\in\mathcal{S}} \|\Delta\mathbf{p}_n - \mathbf{R}[\mathcal{I}_n(\mathbf{p}_n \circ \Delta\mathbf{p}_n) - \mathcal{T}(\mathbf{0})]\|_2^2 + \Omega(\mathbf{R}) \ . \tag{8}$$

The template image $\mathcal{T}(\mathbf{0})$ can be learned either with $\mathbf{R}$ directly or by taking it to be $\frac{1}{N}\sum_{n\in\mathcal{S}}\mathcal{I}(\mathbf{p}_n)$, the average of ground-truth images [15].

Regularization. Ω is a regularization function used to ensure that the solution to $\mathbf{R}$ is unique. To understand the need for this regularization, one can reform Equation 8 in matrix form as

$$\min_{\mathbf{R}} \|\mathbf{Y} - \mathbf{R}\mathbf{X}\|_F^2 + \Omega(\mathbf{R}), \tag{9}$$

where

$$\begin{aligned}\mathbf{Y} &= \left[\Delta\mathbf{p}_1, \ldots, \Delta\mathbf{p}_N\right], \text{and} \\ \mathbf{X} &= \left[\mathcal{I}(\mathbf{p}_1 \circ \Delta\mathbf{p}_1) - \mathcal{T}(\mathbf{0}), \ldots, \mathcal{I}(\mathbf{p}_N \circ \Delta\mathbf{p}_N) - \mathcal{T}(\mathbf{0})\right] \quad .\end{aligned}$$

Here, $\|\cdot\|_F$ indicates the matrix Frobenius norm. Without the regularization term $\Omega(\mathbf{R})$, the solution to Equation 9 is $\mathbf{R} = \mathbf{Y}\mathbf{X}^\top(\mathbf{X}\mathbf{X}^\top)^{-1}$. It is understood in literature that raw pixel representations of natural images stem from certain frequency spectrums [13] that leads to an auto-covariance matrix $\mathbf{X}\mathbf{X}^\top$ which is poorly conditioned in nearly all circumstances. It has been demonstrated [13] that this property stems from the fact that image intensities in natural images are highly correlated in close spatial proximity, but this dependence drops off as a function of spatial distance.

In our experiments, we have found that $\mathbf{X}\mathbf{X}^\top$ is always poorly conditioned even when utilizing other image representations such as dense SIFT, HOG, and LBP descriptors. As such, it is clear that some sort of regularization term is crucial for effective SDM performance. As commonly advocated and practiced, we employed a weighted Tikhonov penalty term $\Omega(\mathbf{R}) = \lambda||\mathbf{R}||_F^2$, where λ controls the weight of the regularizer. We found this choice to work well in our experiments.

Iteration-specific Regressors. Unlike the IC-LK approach, which employs a single regressor/template pair $\{\mathbf{R}, \mathcal{T}(\mathbf{0})\}$ to be applied iteratively until convergence, SDM learns a set of regressor/template pairs $\{\mathbf{R}^{(l)}, \mathcal{T}^{(l)}(\mathbf{0})\}_{l=1}^{L}$ for each iteration $l = 1 : L$ (sometimes referred to as layers). On the other hand, like the IC-LK algorithm, these regressors are precomputed in advance and thus are independent of the current image and warp estimate. As a result, SDM is computationally efficient just like IC-LK. The regressor/template pair $\{\mathbf{R}^{(l)}, \mathcal{T}^{(l)}(\mathbf{0})\}$ is learned from the synthetically generated set $\mathcal{S}^{(l)}$ within Equation 8, which we define to be

$$\mathcal{S}^{(l)} = \{\Delta\mathbf{p}_n^{(l)}, \mathcal{I}(\mathbf{p}_n \circ \Delta\mathbf{p}_n^{(l)})\}_{n=1}^{N}, \tag{10}$$

where

$$\Delta\mathbf{p}^{(l+1)} \leftarrow \mathbf{R}^{(l)}\left[\mathcal{I}\left(\mathbf{p} \circ (\Delta\mathbf{p}^{(l)})^{-1}\right) - \mathcal{T}(\mathbf{0})\right] \quad . \tag{11}$$

For the first iteration ($l = 1$), the warp perturbations are generated from a pre-determined random distribution; for every subsequent iteration, the warp perturbations are re-sampled from the same distribution to ensure each iteration's regressor does not overfit. Once learned, SDM is applied by employing Equation 11 in practice.

Inverse Compositional Warps. It should be noted that there is nothing in the original treatment [14] on SDM that limits it to compositional warps. In fact, the original work employing facial landmark alignment advocated an additive update strategy. In this paper, however, we have chosen to employ inverse compositional warp updates as: (i) we obtained better results for our experiments with planar warp functions, (ii) we observed almost no difference in performance for non-planar warp functions such as those involved in face alignment, and (iii) it is only through the employment of inverse compositional warps within the LK framework that a firm theoretical motivation for fixed regressors can be entertained. Furthermore, we have found that keeping a close mathematical relationship to the IC-LK algorithm is essential for the motivation of our proposed approach.

4 The Conditional Lucas & Kanade Algorithm

Although enjoying impressive results across a myriad of image alignment tasks, SDM does have disadvantages when compared to IC-LK. First, it requires large amounts of synthetically warped image data. Second, it requires the utilization of an adhoc regularization strategy to ensure good condition of the linear system. Third, the mathematical properties of the warp function parameters being predicted is ignored. Finally, it reveals little about the actual degrees of freedom necessary in the set of regressor matrices being learned through the SDM process.

In this paper, we put forward an alternative strategy for directly learning a set of iteration-specific regressors,

$$\min_{\nabla\mathcal{T}(\mathbf{0})} \sum_{n\in\mathcal{S}} \|\Delta\mathbf{p}_n - \mathbf{R}[\mathcal{I}(\mathbf{p}_n \circ \Delta\mathbf{p}_n) - \mathcal{T}(\mathbf{0})]\|_2^2 \tag{12}$$

$$\text{s.t.} \quad \mathbf{R} = \left(\begin{bmatrix} \nabla\mathcal{T}(\mathbf{x}_1) & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \nabla\mathcal{T}(\mathbf{x}_D) \end{bmatrix} \begin{bmatrix} \frac{\partial\mathcal{W}(\mathbf{x}_1;\mathbf{0})}{\partial\mathbf{p}^\top} \\ \vdots \\ \frac{\partial\mathcal{W}(\mathbf{x}_D;\mathbf{0})}{\partial\mathbf{p}^\top} \end{bmatrix} \right)^\dagger,$$

where

$$\nabla\mathcal{T}(\mathbf{0}) = \begin{bmatrix} \nabla\mathcal{T}(\mathbf{x}_1) \\ \vdots \\ \nabla\mathcal{T}(\mathbf{x}_D) \end{bmatrix}.$$

At first glance, this objective may seem strange, as we are proposing to learn template "image gradients" $\nabla\mathcal{T}(\mathbf{0})$ within a conditional objective. As previously discussed in [4], this idea deviates from the traditional view of what image gradients are - parameters that are derived from heuristic finite differencing operations. In this paper, we prefer to subscribe to the alternate view that image gradients are simply weights that can be, and should be, learned from data. The central motivation for this objective is to enforce the parametric form of the generative IC-LK form through a conditional objective.

An advantage of the Conditional LK approach is the reduced number of model parameters. Comparing the model parameters of Conditional LK ($\nabla\mathcal{T}(\mathbf{0}) \in \mathbb{R}^{KD\times 2}$) against SDM ($\mathbf{R} \in \mathbb{R}^{P\times KD}$), there is a reduction in the degrees of freedom needing

to be learned for most warp functions where $P > 2$. More fundamentally, however, is the employment of the generative pixel independence assumption described originally in Equation 1. This independence assumption is useful as it ensures that a unique $\mathbf{R}$ can be found in Equation 12 without any extra penalty terms such as Tikhonov regularization. In fact, we propose that the sparse matrix structure of image gradients within the psuedo-inverse of $\mathbf{R}$ acts as a much more principled form of regularization than those commonly employed within the SDM framework.

A further advantage of our approach is that, like the IC-LK framework, it utilizes prior knowledge of the warp Jacobian function $\frac{\partial \mathcal{W}(\mathbf{x};\mathbf{0})}{\partial \mathbf{p}^\top}$ during the estimation of the regression matrix $\mathbf{R}$. Our insight here is that the estimation of the regression matrix $\mathbf{R}$ using a conditional learning objective should be simplified (in terms of the degrees of freedom to learn) if one had prior knowledge of the deterministic form of the geometric warp function.

A drawback to the approach, in comparison to both the SDM and IC-LK frameworks, is the non-linear form of the objective in Equation 12. This requires us to resort to non-linear optimization methods, which are not as straightforward as linear regression solutions. However, as we discuss in more detail in the experimental portion of this paper, we demonstrate that a Levenberg-Marquardt optimization strategy obtains good results in nearly all circumstances. Furthermore, compared to SDM, we demonstrate good solutions can be obtained with significantly smaller numbers of training samples.

Iteration-specific Regressors. As with SDM, we assume we have an ensemble of images and ground-truth warps $\{\mathcal{I}_m, \mathbf{p}_m\}_{m=1}^M$ from which a much larger set of synthetic examples can be generated $\mathcal{S} = \{\Delta\mathbf{p}_n, \mathcal{I}_n(\mathbf{p}_n \circ \Delta\mathbf{p}_n)\}_{n=1}^N$, where $N \gg M$. Like SDM, we attempt to learn a set of regressor/template pairs $\{\mathbf{R}^{(l)}, \mathcal{T}^{(l)}(\mathbf{0})\}_{l=1}^L$ for each iteration $l = 1 : L$. The set $\mathcal{S}^{(l)}$ of training samples is derived from Equations 10 and 11 for each iteration. Once learned, the application of these iteration-specific regressors is identical to SDM.

Pixel Independence Asymmetry. A major advantage of the IC-LK framework is that it assumes generative independence across pixel coordinates (see Equation 1). A natural question to ask is: could not one predict geometric displacement (instead of appearance) directly across independent pixel coordinates?

The major drawback to employing such strategy is its ignorance of the well-known "aperture problem" [10] in computer vision (*e.g.* the motion of an image patch containing a sole edge cannot be uniquely determined due to the ambiguity of motion along the edge). As such, it is impossible to ask any predictor (linear or otherwise) to determine the geometric displacement of all pixels within an image while entertaining an independence assumption. The essence of our proposed approach is that it circumvents this issue by enforcing global knowledge of the template's appearance across all pixel coordinates, while entertaining the generative pixel independence assumption that has served the LK algorithm so well over the last three decades.

Generative LK. For completeness, we will also entertain a generative form of our objective in Equation 12, where we instead learn "image gradients" that predict generative

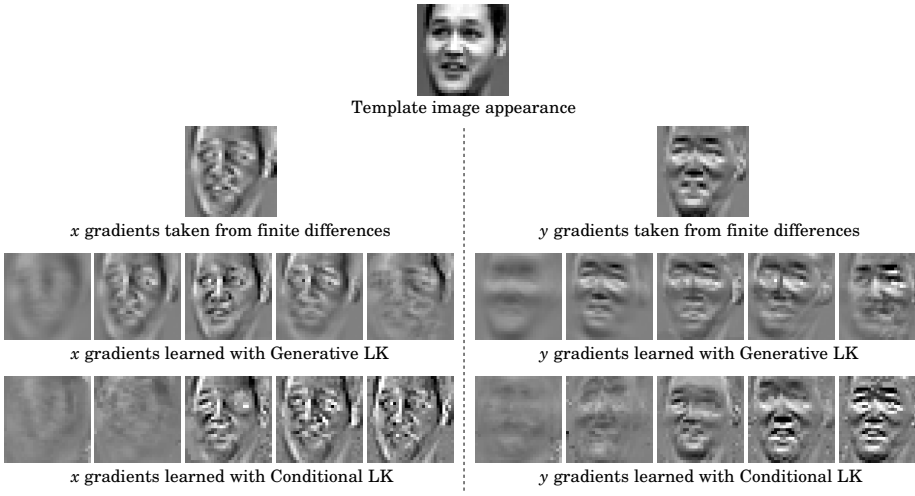


Fig. 1. Visualization of the learned image gradients for LK from layers 1 (left) to 5 (right).

appearance as a function of geometric displacement, formulated as

$$\min_{\nabla\mathcal{T}(\mathbf{0})} \sum_{n\in\mathcal{S}} \|\mathcal{I}(\mathbf{p}_n \circ \Delta\mathbf{p}_n) - \mathcal{T}(\mathbf{0}) - \mathbf{W}\Delta\mathbf{p}_n\|_2^2 \tag{13}$$

$$\text{s.t.} \quad \mathbf{W} = \begin{bmatrix} \nabla\mathcal{T}(\mathbf{x}_1) & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \nabla\mathcal{T}(\mathbf{x}_D) \end{bmatrix} \begin{bmatrix} \frac{\partial\mathcal{W}(\mathbf{x}_1;\mathbf{0})}{\partial\mathbf{p}^\top} \\ \vdots \\ \frac{\partial\mathcal{W}(\mathbf{x}_D;\mathbf{0})}{\partial\mathbf{p}^\top} \end{bmatrix}.$$

Unlike our proposed Conditional LK, the objective in Equation 13 is linear and directly solvable. Furthermore, due to the generative pixel independence assumption, the problem can be broken down into D independent sub-problems. The Generative LK approach is trained in an identical way to SDM and Conditional LK, where iteration-specific regressors are learned from a set of synthetic examples $\mathcal{S} = \{\Delta\mathbf{p}_n, \mathcal{I}_n(\mathbf{p}_n \circ \Delta\mathbf{p}_n)\}_{n=1}^N$.

Figure 1 provides an example of visualizing the gradients learned from the Conditional LK and Generative LK approaches. It is worthwhile to note that the Conditional LK gradients get sharper over regression iterations, while it is not necessarily the case for Generative LK. The rationale for including the Generative LK form is to highlight the importance of a conditional learning approach, and to therefore justify the added non-linear complexity of the objective in Equation 12.

5 Experiments

In this section, we present results for our approach across three diverse tasks: (i) planar image alignment, (ii) planar template tracking, and (iii) facial model fitting. We also investigate the utility of our approach across different image representations such as raw pixel intensities and dense LBP descriptors.

Fig. 2. Visualization of the perturbed samples $\mathcal{S} = \{\Delta\mathbf{p}_n, \mathcal{I}_n(\mathbf{p}_n \circ \Delta\mathbf{p}_n)\}_{n=1}^N$ used for training the SDM, Conditional LK, and Generative LK methods. Left: the original source image, where the red box is the ground truth and the green boxes are perturbed for training. Right: examples of the synthesized training samples.

5.1 Planar Image Alignment

Experimental settings. In this portion of our experiments, we will be utilizing a subsection of the Multi-PIE [6] dataset. For each image, we denote a 20×20 image $\mathcal{I}(\mathbf{p})$ with ground-truth warp $\mathbf{p}$ rotated, scaled and translated around hand-labeled locations. For the IC-LK approach, this image is then employed as the template $\mathcal{T}(\mathbf{0})$. For the SDM, Conditional LK and Generative LK methods, a synthetic set of geometrically perturbed samples $\mathcal{S}$ are generated $\mathcal{S} = \{\Delta\mathbf{p}_n, \mathcal{I}_n(\mathbf{p}_n \circ \Delta\mathbf{p}_n)\}_{n=1}^N$.

We generate the perturbed samples by adding i.i.d. Gaussian noise of standard deviation σ to the four corners of the ground-truth bounding box as well as an additional translational noise from the same distribution, and then finally fitting the perturbed box to the warp parameters $\Delta\mathbf{p}$. In our experiments, we choose $\sigma = 1.2$ pixels. Figure 2 shows an example visualization of the training procedure as well as the generated samples. For SDM, a Tikhonov regularization term is added to the training objective as described in Section 3, and the penalty factor λ is chosen by evaluating on a separate validation set; for Conditional LK, we use Levenberg-Marquardt to optimize the non-linear objective where the parameters are initialized through the Generative LK solution.

Frequency of Convergence. We compare the alignment performance of the four types of aligners in our discussion: (i) IC-LK, (ii) SDM, (iii) Generative LK, and (iv) Conditional LK. We state that convergence is reached when the point RMSE of the four corners of the bounding box is less than one pixel.

Figure 3 shows the frequency of convergence tested with both a 2D affine and homography warp function. Irrespective of the planar warping function, our results indicate that Conditional LK has superior convergence properties over the others. This result holds even when the approach is initialized with a warp perturbation that is larger than the distribution it was trained under. The alignment performance of Conditional

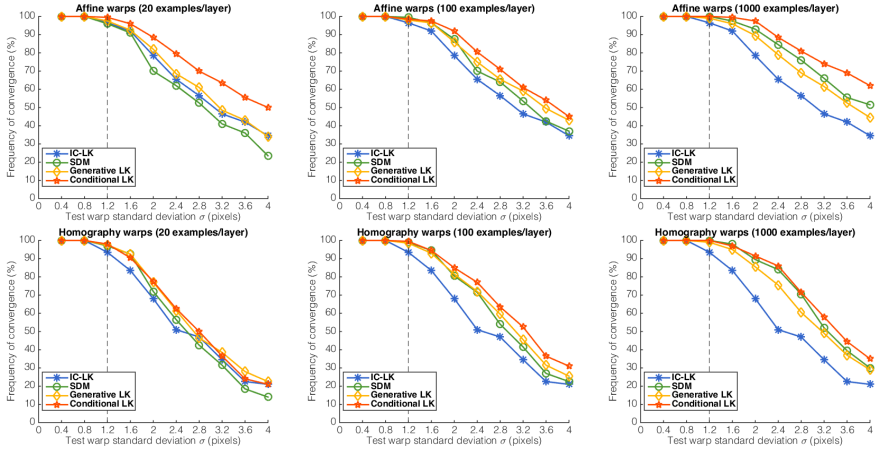


Fig. 3. Frequency of convergence comparison between IC-LK, SDM, Generative LK, and Conditional LK. The vertical dotted line indicates σ that they were trained with.

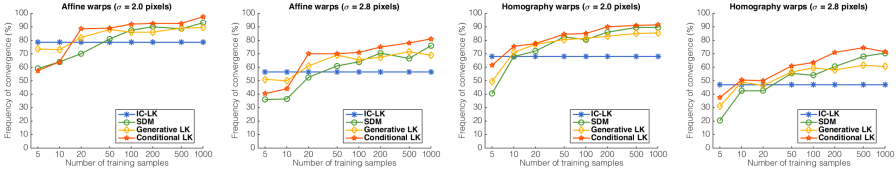


Fig. 4. Frequency of convergence comparison between SDM, Generative LK, and Conditional LK in terms of number of samples trained with.

LK is consistently better in all circumstances, although the advantage of the approach is most noticeable when training with just a few training samples.

Figure 4 provides another comparison with respect to the amount of training data learned from. It can be observed that SDM is highly dependent on the amount of training data available, but it is still not able to generalize as well as Conditional LK. This is also empirical proof that incorporating principled priors in Conditional LK is more desirable than adhoc regularizations in SDM.

Convergence Rate. We also provide some analysis on the convergence speed. To make a fair comparison, we take the average of only those test runs where all regressors converged. Figure 5 illustrates the convergence rates of different regressors learned from different amounts of training data. The improvement of Conditional LK in convergence speed is clear, especially when little training data is provided. SDM starts to exhibit faster convergence rate when learned from over 100 examples per layer; however, Conditional LK still surpasses SDM in term of the frequency of final convergence.

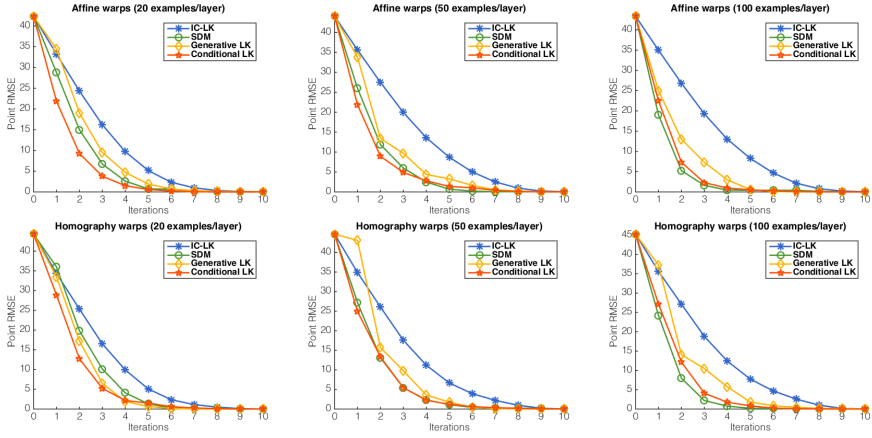


Fig. 5. Convergence rate comparison between IC-LK, SDM, Generative LK, and Conditional LK, averaged from the tests ($\sigma = 2.8$) where all four converged in the end.

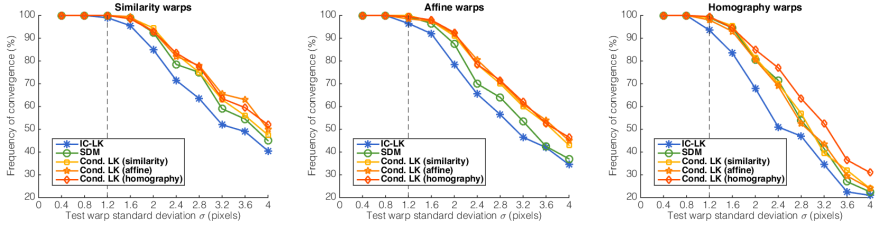


Fig. 6. Frequency of convergence comparison between IC-LK, SDM, and Conditional LK trained with 100 examples per layer and tested with swapped warp functions. The parentheses indicate the type of warp function trained with.

Swapping Warp Functions. A unique property of Conditional LK in relation to SDM is its ability to interchange between warp functions after training. Since we are learning image gradients $\nabla\mathcal{T}(\mathbf{0})$ for the Conditional LK algorithm, one can essentially choose which warp Jacobian to be employed before forming the regressor $\mathbf{R}$. Figure 6 illustrates the effect of Conditional LK learning the gradient with one type of warp function and swapping it with another during testing. We see that whichever warp function Conditional LK is learned with, the learned conditional gradients are also effective on the other and still outperforms IC-LK and SDM.

It is interesting to note that when we learn the Conditional LK gradients using either 2D planar similarity warps ($P = 4$) or homography warps ($P = 8$), the performance on 2D planar affine warps ($P = 6$) is as effective. This outcome leads to an important insight: it is possible to learn the conditional gradients with a simple warp function and replace it with a more complex one afterwards; this can be especially useful when certain types of warp functions (*e.g.* 3D warp functions) are harder to come by.

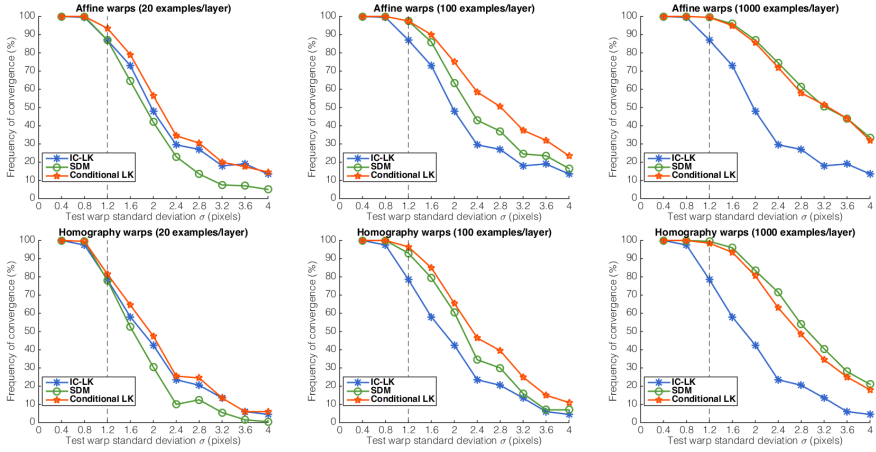


Fig. 7. Frequency of convergence comparison between IC-LK, SDM and Conditional LK with dense binary descriptors. The vertical dotted line indicates σ that they were trained with.

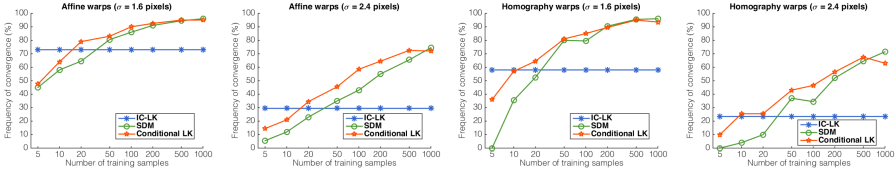


Fig. 8. Frequency of convergence comparison between SDM and Conditional LK with dense binary descriptors in terms of number of samples trained with.

5.2 Planar Tracking with LBP Features

In this section, we show how Conditional LK can be effectively employed with dense multi-channel LBP descriptors where $K = 8$. First we analyze the convergence properties of Conditional LK on the dense LBP descriptors, as we did similarly in the previous section, and then we present an application to robust planar tracking. A full description of the multi-channel LBP descriptors we used in our approach can be found in [1].

Figure 7 provides a comparison of robustness by evaluating the frequency of convergence with respect to the scale of test warps σ. This suggests that Conditional LK is as effective in the LK framework with multi-channel descriptors: in addition to increasing alignment robustness (which is already a well-understood property of descriptor image alignment), Conditional LK is able to improve upon the sensitivity to initialization with larger warps.

Figure 8 illustrates alignment performance as a function of the number of samples used in training. We can see the Conditional LK only requires as few as 20 examples per layer to train a better multi-channel aligner than IC-LK, whereas SDM needs more than 50 examples per iteration-specific regressor. This result again speaks to the efficiency of learning with Conditional LK.

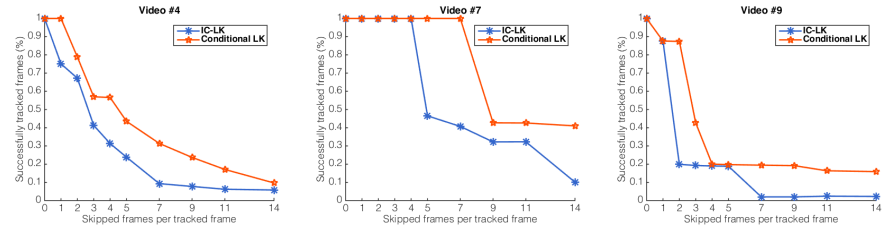


Fig. 9. Tracking performance using IC-LK and Conditional LK with dense LBP descriptors for three videos under low frame-rate conditions, with and without lighting variations.

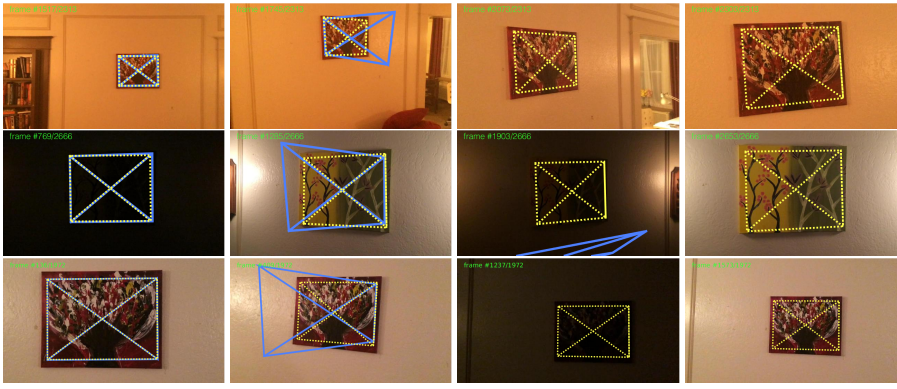
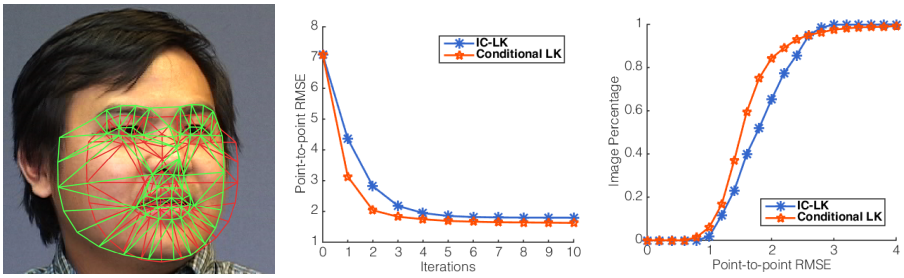


Fig. 10. Snapshots of tracking results. Blue: IC-LK; yellow: Conditional LK. The second image of each row shows where IC-LK fails but Conditional LK still holds.

Low Frame-rate Template Tracking. In this experiment, we evaluate the advantage of our proposed approach for the task of low frame-rate template tracking. Specifically, we borrow a similar experimental setup to Bit-Planes [1]. LBP-style dense descriptors are ideal for this type of task as their computation is computationally feasible in real-time across a number of computational platforms (unlike HOG or dense SIFT). Further computational speedups can be entertained if we start to skip frames to track.

We compare the performance of Conditional LK with IC-LK and run the experiments on the videos collected in [1]. We train the Conditional LK tracker on the first frame with 20 synthetic examples. During tracking, we skip every k frames to simulate low frame-rate videos. Figure 9 illustrates the percentage of successfully tracked frames over the number of skipped frames k. It is clear that the Conditional LK tracker is more stable and tolerant to larger displacements between frames.

Figure 10 shows some snapshots of the video, including the frames where the IC-LK tracker starts to fail but the Conditional LK tracker remains. This further demonstrates that the Conditional LK tracker maintains the same robustness to brightness variations by entertaining dense descriptors, but meanwhile improves upon convergence. Enhanced susceptibility to noises both in motion and brightness also suggests possible extensions to a wide variety of tracking applications.



(a) (b) (c)

Fig. 11. (a) An example of facial model fitting. The red shape indicates the initialization, and the green shape is the final fitting result. (b) Convergence rate comparison between IC-LK and Conditional LK. (c) Comparison of fitting accuracy.

5.3 Facial Model Fitting

In this experiment, we show how Conditional LK is applicable not only to 2D planar warps like affine or homography, but also to more complex warps that requires heavier parametrization. Specifically, we investigate the performance of our approach with a point distribution model (PDM) [11] on the IJAGS dataset [11], which contains an assortment of videos with hand-labeled facial landmarks. We utilize a pretrained 2D PDM learned from all labeled data as the warp Jacobian and compare the Conditional LK approach against IC-LK (it has been shown that there is an IC formulation to facial model fitting [11]). For Conditional LK, we learn a series of regressor/template pairs with 5 examples per layer; for IC-LK, the template image is taken by the mean appearance.

Figure 11 shows the results of fitting accuracy and convergence rate of subject-specific alignment measured in terms of the point-to-point RMSE of the facial landmarks; it is clear that Conditional LK outperforms IC-LK in convergence speed and fitting accuracy. This experiment highlights the possibility of extending our proposed Conditional LK to more sophisticated warps. We would like to note that it is possible to take advantage of the Conditional LK warp swapping property to incorporate a 3D PDM as to introduce 3D shape modelling; this is beyond the scope of discussion of this paper.

6 Conclusion

In this paper, we discuss the advantages and drawbacks of the LK algorithm in comparison to SDMs. We argue that by enforcing the pixel independence assumption into a conditional learning strategy we can devise a method that: (i) utilizes substantially less training examples, (ii) offers a principled strategy for regularization, and (iii) offers unique properties for adapting and modifying the warp function after learning. Experimental results demonstrate that the Conditional LK algorithm outperforms both the LK and SDM algorithms in terms of convergence. We also demonstrate that Conditional LK can be integrated with a variety of applications that potentially leads to other exciting avenues for investigation.

References

1. Alismail, H., Browning, B., Lucey, S.: Bit-planes: Dense subpixel alignment of binary descriptors. CoRR abs/1602.00307 (2016), http://arxiv.org/abs/1602.00307
2. Antonakos, E., Alabort-i Medina, J., Tzimiropoulos, G., Zafeiriou, S.P.: Feature-based lucas–kanade and active appearance models. Image Processing, IEEE Transactions on 24(9), 2617–2632 (2015)
3. Baker, S., Matthews, I.: Lucas-kanade 20 years on: A unifying framework. International journal of computer vision 56(3), 221–255 (2004)
4. Bristow, H., Lucey, S.: In defense of gradient-based alignment on densely sampled sparse features. In: Dense correspondences in computer vision. Springer (2014)
5. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on. vol. 1, pp. 886–893. IEEE (2005)
6. Gross, R., Matthews, I., Cohn, J., Kanade, T., Baker, S.: Multi-pie. Image and Vision Computing 28(5), 807–813 (2010)
7. Jebara, T.: Discriminative, generative and imitative learning. Ph.D. thesis, Massachusetts Institute of Technology (2001)
8. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. International journal of computer vision 60(2), 91–110 (2004)
9. Lucas, B.D., Kanade, T., et al.: An iterative image registration technique with an application to stereo vision. In: IJCAI. vol. 81, pp. 674–679 (1981)
10. Marr, D.: Vision: A computational investigation into the human representation and processing of visual information, henry holt and co. Inc., New York, NY 2 (1982)
11. Matthews, I., Baker, S.: Active appearance models revisited. International Journal of Computer Vision 60(2), 135–164 (2004)
12. Ojala, T., Pietikäinen, M., Mäenpää, T.: Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. Pattern Analysis and Machine Intelligence, IEEE Transactions on 24(7), 971–987 (2002)
13. Simoncelli, E.P., Olshausen, B.A.: Natural image statistics and neural representation. Annual review of neuroscience 24(1), 1193–1216 (2001)
14. Xiong, X., De la Torre, F.: Supervised descent method and its applications to face alignment. In: Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on. pp. 532–539. IEEE (2013)
15. Xiong, X., De la Torre, F.: Global supervised descent method. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2664–2673 (2015)
16. Xiong, X., la Torre, F.D.: Supervised descent method for solving nonlinear least squares problems in computer vision. CoRR abs/1405.0601 (2014), http://arxiv.org/abs/1405.0601