Registration based Tracking with Structural Similarity: A Unifying Formulation

Abhineet Singh and Martin Jagersand
University of Alberta

Abstract. This paper adapts a popular image quality measure called Structural Similarity for high precision registration based tracking while also introducing a simpler and faster variant of the same. Further, to evaluate these comprehensively against existing measures, it presents a unified way to study registration based trackers by decomposing them into three constituent sub modules: appearance model, state space model and search method. This approach has relevance beyond the present work as it is often the case that when a new tracker is introduced in this domain, it only contributes to one or two of these sub modules while using existing methods for the rest. Since these are often selected arbitrarily by the authors, they may not be optimal for the new method. In such cases, this breakdown can help to experimentally find the best combination of methods for these sub modules while also providing a framework within which the contributions of the new tracker can be clearly demarcated and thus studied better. All experiments are performed using an open source tracking framework on three publicly available datasets so the results are easily reproducible. In addition, this framework, by following this decomposition closely through extensive use of generic programming, provides a convenient interface to plug in a new method for any sub module and combine it with existing methods for the others. It can also serve as a fast and flexible solution for practical tracking requirements owing to its highly efficient C++ implementation.

1 Introduction

Since its inception, research in object tracking has focused on presenting new algorithms to address specific challenges in a wide variety of application domains like robotics, surveillance, targeting systems, augmented reality and medical analysis. However, before a tracker can be adopted in a real life application, it needs to be extensively tested so that both its advantages and limitations can be determined. Recent studies in tracking evaluation [1,2] show increasing efforts to standardize this crucial process. However, though such studies assign a global rank to each tracker, they often provide little feedback to improve these since they treat them as black boxes predicting the trajectory of the object. A more useful evaluation methodology would be to have empirical validation of the tracker’s design or point out its shortcomings. An exhaustive analysis of learning based trackers is admittedly a daunting and impracticable task as these often use widely varying techniques that have little in common.

This, however, is not true for registration based trackers [3,4]. As we show in Sec. 2.2, these can be decomposed into three well defined modules - appearance model
(AM), state space model (SSM) and search method (SM) - thus making their systematic analysis feasible. Many reported studies in this domain [3,4,5] have introduced new methods for only one of these submodules without exploring the full extent of their contributions. For instance, Baker et. al [4] reported a compositional update scheme for the SSM parameters instead of the additive scheme used in [6], but only tested it with SSD. Conversely, Richa et. al [7] showed an improvement over the existing ESM [5] approach by using SCV as the AM (Sec. 2.2) instead of SSD. Similarly, Dame et. al [8] used MI while Scandaroli et. al [9] used NCC with the ICLK SM [4]. However, in none of these works were the respective AMs tested with other SMs. There are two steps in finding the optimal combination of methods for a tracking algorithm - determine the sub module where the algorithm’s main contribution lies and evaluate all possible combinations of methods for the other sub modules that are compatible with it. A generic framework would thus be useful to avoid such fragmentation.

We demonstrate the practical applicability of this approach by comparing several existing methods for these sub modules not only with each other but also with two new AMs based on structural similarity (SSIM) [10] that we introduce and fit within this framework. SSIM is a popular measure for evaluating the quality of image compression algorithms by comparing the compressed image with the original one. Since it measures the information loss in the former - essentially a slightly distorted or damaged version of the latter - it makes a suitable metric for comparing candidate warped patches with the template to find the one with the minimum loss and thus most likely to represent the same object. Further, it has been designed to capture the perceptual similarity of images and is known to be robust to illumination and contrast changes [10]. It should, therefore, perform well for tracking under challenging conditions. As such, it has indeed been used for tracking before with particle filters [11,12], gradient ascent [13] and hybrid [14] approaches. All of these trackers, however, used imprecise SSMs with low degrees of freedom (DOF) - estimating only translation and scaling of the target patch. To the best of our knowledge, no attempt has been made to use SSIM for high DOF registration based tracking within the Lucas Kanade framework [3]. This work aims to fill this gap.

To summarize, following are the main contributions of this work:

- Apply a popular image quality measure - SSIM - for high precision registration based tracking and introduce a simpler but faster version called SPSS (Sec. 3).
- Evaluate these models comprehensively by comparing against 8 existing AMs using 10 SMs and 7 SSMs. Experiments are done using three large datasets with almost 90,000 frames in all to ensure their statistical significance.
- Report for the first time, to the best of our knowledge, results comparing robust similarity metrics [15] with traditional SSD type measures.
- Compare formulations against state of the art online learning based trackers to validate their usability in precise tracking applications.
- Provide an open source tracking framework [16] using which all results can be reproduced and which, owing to its efficient C++ implementation, can also be used to address practical tracking requirements.
2 Background

2.1 Notation

Let $I_t : \mathbb{R}^2 \mapsto \mathbb{R}$ refer to an image captured at time $t$ so that a video stream can be modeled as a sequence of images $\{I_t| t \geq 0\}$. The pixel locations in an image patch with $N$ pixels are denoted by a $2 \times N$ matrix $\mathbf{x} = [x^1, x^2, \ldots, x^N]$ where $x^k = [x^k, y^k]^T \in \mathbb{R}^2$ are the Cartesian coordinates of pixel $k$ in image space. The corresponding pixel intensities in image $I$ are represented by an $N \times 1$ vector $\mathbf{I}(\mathbf{x}) = [I(x^1, y^1), I(x^2, y^2), \ldots, I(x^N, y^N)]^T \in \mathbb{R}^N$. $\mathbf{I}(\mathbf{x})$ may also be referred to simply as $I$ when the meaning is clear from context. Additionally, let $\mathbf{x}_t$ denote the patch corresponding to the tracked object’s location in $I_t$ where $x_0$ is specified manually and estimating the rest is the goal of the tracking task.

Further, let $\mathbf{w}(\mathbf{x}, \mathbf{p}) : \mathbb{R}^2 \times \mathbb{R}^S \mapsto \mathbb{R}^2$ denote a warping function of $S$ parameters $\mathbf{p} = (p^1, p^2, \ldots, p^S)$ that represents the set of allowable image motions of the tracked object by specifying the deformations that can be applied to $\mathbf{x}_0$ to align it with $\mathbf{x}_t = \mathbf{w}(\mathbf{x}_0, \mathbf{p}_t)$. Finally let $f(\mathbf{I}^*, \mathbf{I}^c) : \mathbb{R}^N \times \mathbb{R}^N \mapsto \mathbb{R}$ be a function that measures the similarity between two sets of pixel values: the reference or template patch $\mathbf{I}^*$, typically extracted from $I_0$, and a candidate patch $\mathbf{I}^c$ extracted from $I_t$.

2.2 Decomposing registration based tracking

With the above notation, registration based tracking can be formulated (Eq 1) as a search problem where the goal is to find the optimal warping parameters $\mathbf{p}_t$ for an image $I_t$ that maximize the similarity, measured by $f$, between the target patch $\mathbf{I}^* = I_0(\mathbf{w}(\mathbf{x}_0, \mathbf{p}_0)) = I_0(\mathbf{x}_0)$ and the warped image patch $\mathbf{I}^c = I_t(\mathbf{w}(\mathbf{x}_0, \mathbf{p}_t)) = I_t(\mathbf{x}_t)$.

$$\mathbf{p}_t = \arg\max_{\mathbf{p}} f(I_0(\mathbf{x}_0), I_t(\mathbf{w}(\mathbf{x}_0, \mathbf{p})))$$

As has been observed before [17,15], this formulation gives rise to an intuitive way to decompose the tracking task into three modules - the similarity metric $f$, the warping function $\mathbf{w}$ and the optimization approach. These can be designed to be semi independent in the sense that any given optimizer can be applied unchanged to several combinations of methods for the other two modules which in turn interact only through a well defined and consistent interface. In this work, we refer to these modules respectively as appearance model ($\text{AM}$), state space model ($\text{SSM}$) and search method ($\text{SM}$). A more detailed description of these sub modules follows.

**Search Method** This is the optimization procedure that searches for the warped patch in $I_t$ that best matches $I_0(\mathbf{x}_0)$. Gradient descent optimization is the most popular approach used in tracking due to its speed and simplicity and is the basis for the classic Lucas Kanade (LK) tracker [3]. This algorithm can be formulated in four different ways [4] and the resulting variants - forward additive ($\text{FALK}$) [3], inverse additive ($\text{IALK}$) [6], forward compositional ($\text{FCLK}$) [17] and inverse compositional ($\text{ICLK}$) [4] - were analyzed mathematically and shown to be equivalent to first order terms in [4]. Here, however, we show experimental results proving that their performance on real benchmarks is quite different (Sec. 4.4). A relatively recent update to this approach was in the
form of Efficient Second order Minimization (ESM) [5] technique that tries to make the best of both inverse and forward formulations by using information from both the initial and current Jacobians.

Nearest neighbor search (NN) is a stochastic SM that has recently been used for tracking [18] thanks to the FLANN library [19] that makes real time search feasible. Since the performance of stochastic SMs like NN depends largely on the number of random samples used, we have reported results with 1000 and 10000 samples, with the respective SMs named as NN1K and NN10K. Further, NN tends to give jittery and unstable results when used by itself due to the very limited search space and so was used in conjunction with a gradient descent type SM in [18] to create a composite tracker that performs better than either of its constituents. As in [18], we have used ICLK as this second tracker due to its speed and the resultant composite SM is named NNIC.

Another recent stochastic SM is RKLT [20] that samples the target patch into a regularly spaced grid of smaller sub patches, tracks each sub patch over two consecutive frames using a 2 DOF tracker and finally uses RANSAC to estimate SSM parameters that can best explain the corresponding positions of these trackers in the two frames. Similarly to NN, this method too tends to be unstable and imprecise when used by itself, so is combined with a few iterations of ICLK. We refer to the first stochastic stage of this composite SM as RANSAC and the overall SM (RANSAC + ICLK) as RKLT.

**Appearance Model** This is the similarity metric defined by the function $f$ in Eq. 1 using which the SM compares different warped patches from the current image to get the closest match with the original template. The sum of squared differences (SSD) [3,4,5] is the AM most often used in literature especially with SMs based on gradient descent search due to its simplicity and the ease of computing its derivatives. However, the same simplicity also makes it vulnerable to providing false matches when the object’s appearance changes due to factors like illumination variations or motion blur.

To address these issues, more robust AMs have been proposed including Sum of Conditional Variance (SCV) [7], Normalized Cross Correlation (NCC) [21], Mutual Information (MI) [22,8] and Cross Cumulative Residual Entropy (CCRE) [23,15], all of which supposedly provide a degree of invariance to changes in illumination. There also exists a slightly different formulation of SCV known as Reversed SCV (RSCV) [18] where $I_t$ is updated rather than $I_0$. There has also been a recent extension to it called LSCV [24] that uses multiple joint histograms from corresponding sub regions within the target patch to achieve greater robustness to localized intensity changes. Finally, it has been shown [25] that maximizing NCC between two images is equivalent to minimizing the SSD between two z-score [26] normalized images. We consider the resultant formulation as a different AM called Zero Mean NCC (ZNCC).

**State Space Model** This represents the set of allowable image motions of the tracked object and thus embodies any constraints that are placed on the search space of warp parameters to make the optimization more efficient. This includes both the degrees of freedom (DOF) of allowed motion, as well as the actual parameterization of the warping function. For instance the ESM tracker, as presented in [5], can be considered to have a different SSM than conventional LK type trackers [3,4] even though both involve 8
DOF homography, since it uses the \( \mathbb{S}L(3) \) parameterization rather than the actual entries of the corresponding matrix. We model 7 different SSMs including 5 from the standard hierarchy of geometrical transformations [27,17] - translation, isometry, similarity, affine and homography - along with two extra parameterizations of homography - \( \mathbb{S}L(3) \) and corner based (using x,y coordinates of the four corners of the bounding box).

The advantage of using higher DOF SSM is achieving greater precision in the aligned warp since transforms that are higher up in the hierarchy [27] can better approximate the projective transformation process that captures the relative motion between the camera and the object in the 3D world into the 2D images. However, there are two issues with having to estimate more parameters - the iterative search takes longer to converge making the tracker slower and the search process becomes more likely to either diverge or end up in a local optimum causing the tracker to be less stable and more likely to lose track. The latter is a well known phenomenon with LK type trackers [28] whose higher DOF variants are usually less robust.

### 3 Structural Similarity

SSIM was originally introduced [10] to assess the loss in image quality incurred by compression methods like JPEG. It has been very popular in this domain since it closely mirrors the approach adopted by the human visual system to subjectively evaluate the quality of an image. The SSIM between two image patches \( I_0 \) and \( I_t \) is defined as a product of 3 components:

\[
    f_{ssim} = \left( \frac{2 \mu_t \mu_0 + C_1}{\mu_t^2 + \mu_0^2 + C_1} \right)^\alpha \left( \frac{2 \sigma_t \sigma_0 + C_2}{\sigma_t^2 + \sigma_0^2 + C_2} \right)^\beta \left( \frac{\sigma_{t0} + C_3}{\sigma_t \sigma_0 + C_3} \right)^\gamma
\]

(2)

where \( \mu_t = \frac{1}{N} \sum_{i=1}^{N} I_t(x_t^i) \) is the mean and \( \sigma_t = \sqrt{\frac{1}{N-1} \sum_{i=1}^{N} (I_t(x_t^i) - \mu_t)^2} \) is the sample standard deviation of intensity values in \( I_t(x_t) \) while \( \sigma_{t0} = \frac{1}{N-1} \sum_{i=1}^{N} (I_t(x_t^i) - \mu_t)(I_0(x_0^i) - \mu_0) \) is the sample covariance between \( I_t(x_t) \) and \( I_0(x_0) \). The three components of \( f_{ssim} \) from left to right are respectively used for luminance, contrast and structure comparison between the two patches. The positive constants \( \alpha, \beta, \gamma \) are used to assign relative weights to these components while \( C_1, C_2, C_3 \) are added to ensure their numerical stability with small denominators. Here, as in most practical implementations [10,11,14,13,12], it is assumed that \( \alpha = \beta = \gamma = 1 \) and \( C_3 = \frac{C_2}{2} \) so that Eq. 2 simplifies to:

\[
    f_{ssim} = \frac{(2 \mu_t \mu_0 + C_1)(2 \sigma_{t0} + C_2)}{(\mu_t^2 + \mu_0^2 + C_1)(\sigma_t^2 + \sigma_0^2 + C_2)}
\]

(3)
3.1 Newton’s Method with SSIM

Using SSIM with NN is quite straightforward since it only needs $f_{ssim}$ to be computed between candidate patches. However, gradient descent based SMs including ESM and the four variants of LK also require its derivatives as they solve Eq 1 by estimating an incremental update $\Delta p_t$ to the optimal SSM parameters $p_{t-1}$ at time $t-1$ using some variant of the Newton method as:

$$\Delta p_t = -\hat{H}^{-1}\hat{J}^T$$

where $\hat{J}$ and $\hat{H}$ respectively are estimates for the $1 \times S$ Jacobian $J = \frac{\partial f}{\partial p}$ and the $S \times S$ Hessian $H = \frac{\partial^2 f}{\partial p^2}$ of the AM w.r.t. SSM parameters. These can be further decomposed using chain rule as:

$$J = \frac{\partial f}{\partial I} \frac{\partial I}{\partial p} = \frac{\partial f}{\partial I} \nabla I \frac{\partial w}{\partial p}$$

$$H = \frac{\partial I^T}{\partial p} \frac{\partial^2 f}{\partial I \partial p} + \frac{\partial f}{\partial I} \frac{\partial^2 I}{\partial p}$$

Of the various terms on the right hand sides of Eqs. 5 and 6, only $\frac{\partial f}{\partial I}$ and $\frac{\partial^2 f}{\partial I^2}$ depend on $f$ so the relevant expressions for $f_{ssim}$ are presented next - general formulations corresponding to $J$ and $H$ are given first, followed by specializations for $\hat{J}$ and $\hat{H}$ used by specific SMs.

SSIM Jacobian For clarity and brevity in the subsequent expressions, we first express SSIM in a simplified form as:

$$f_{ssim} = \frac{ab}{cd}$$

with $a = 2\mu_t\mu_0 + C_1$, $b = 2\sigma_{t0} + C_2$, $c = \mu_t^2 + \mu_0^2 + C_1$ and $d = \sigma_t^2 + \sigma_0^2 + C_2$. We further let $\bar{I}_t$ refer to a mean normalized version of $I_t$ so that $\bar{I}_t = I_t - \mu_t$ and $\sum_{i=1}^{N} \bar{I}_t(x_t^i) = 0$. Differentiating Eq. 7 w.r.t. $I_t$, we get:

$$\frac{\partial f_{ssim}}{\partial I_t} = \frac{1}{cd} [(a'b + b'a) - f_{ssim}(c'd + d'c)]$$

with

$$a' = \frac{\partial a}{\partial I_t} = \frac{2\mu_0}{N} \sum_{i=1}^{N} \bar{I}_t(x_t^i) = \frac{2\mu_0}{N} 1_N$$

$$b' = \frac{\partial b}{\partial I_t} = \frac{2}{N-1} \frac{\partial}{\partial I_t} \sum_{i=1}^{N} \bar{I}_t(x_t^i) \bar{I}_0(x_0^i) = \frac{2\bar{I}_0}{N-1}$$
\[
c' = \frac{\partial c}{\partial I_t} = \frac{2\mu_t}{N} \frac{\partial \sum_{i=1}^N I_t(x_i^t)}{\partial I_t} = \frac{2\mu_t}{N} 1_N \tag{11}
\]
\[
d' = \frac{\partial d}{\partial I_t} = \frac{1}{N-1} \frac{\partial \sum_{i=1}^N (\bar{I}_t(x_i^t))^2}{\partial I_t} = \frac{2\bar{I}_t}{N-1} \tag{12}
\]

where \(1_N\) denotes a \(1\times N\) vector of all ones. The last equality in Eq. 10 follows since
\[
\forall j \in \{1..N\}, \frac{\partial \sum_{i=1}^N \bar{I}_t(x_i^t)}{\partial I_t} = \bar{I}_0(x_j^t) - \frac{1}{N} \sum_{i=1}^N \bar{I}_0(x_i^0) = \bar{I}_0(x_j^t).
\]
Similar reasoning holds for Eq. 12 too. Substituting Eqs. 9 - 12 in Eq. 8 gives:
\[
\frac{\partial f_{ssim}}{\partial I_t} = \frac{2}{cd} \left[ \left( \frac{\mu_0 b}{N} 1_N + \frac{a\bar{I}_0}{N-1} \right) - f_{ssim} \left( \frac{\mu_t d}{N} 1_N + \frac{c\bar{I}_t}{N-1} \right) \right] \tag{13}
\]

**SSIM Hessian** Referring to \(f_{ssim}\) as \(f\) and \(\frac{\partial f_{ssim}}{\partial I_t}\) as \(f'\) for brevity and letting
\[
A = \frac{b\mu_0 - fd\mu_t}{N} 1_N, \quad B = \frac{a\bar{I}_0 - fc\bar{I}_t}{N-1} \quad \text{and} \quad D = \frac{cd}{2},
\]
Eq. 13 can be rewritten as:
\[
f' = \frac{A + B}{D} \tag{14}
\]
Differentiating Eq. 14 w.r.t. \(I_t\), we get:
\[
\frac{\partial^2 f}{\partial I_t^2} = \frac{1}{D} \left( (A' + B') - f'^T D' \right) \tag{15}
\]
with
\[
A' = \frac{\partial A}{\partial I_t} = S_N \left( \frac{1}{N} \left[ \mu_0 b' - \mu_t (df' + fd') - \frac{fd}{N} 1_N \right] \right) \tag{16}
\]
\[
B' = \frac{\partial B}{\partial I_t} = \frac{1}{N-1} (\bar{I}_0 a' - \bar{I}_t (cf' + fc') - \text{diag} (fc 1_N)) \tag{17}
\]
\[
D' = \frac{\partial D}{\partial I_t} = \frac{dc' + cd'}{2} = \frac{1}{2} \left( \frac{\mu_t d}{N} 1_N + \frac{c\bar{I}_t}{N-1} \right) \tag{18}
\]

where \(S_n(K)\) denotes an \(n \times k\) matrix formed by stacking the \(1 \times k\) vector \(K\) into rows while \(\text{diag}(K)\) denotes a \(k \times k\) diagonal matrix with \(K\) as the principal diagonal.

**Formulations for specific methods** The form of \(\frac{\partial f}{\partial I}\) used by the four variants of LK is identical except that ICLK requires the differentiation to be done w.r.t. \(I_0\) instead of \(I_t\) - the expressions for this are trivial to derive since SSIM is symmetrical. In its original paper [5], ESM was formulated as using the mean of the image gradients \(\nabla I_0\) and \(\nabla I_t\) to compute \(J\) but, as this formulation is only applicable to SSD, we consider a generalized version [29,21] that uses the difference between FCLK and ICLK Jacobians. The reader is referred to [4] and [16] for more details.
It is generally assumed [4,5] that the second term of Eq. 6 is too costly to compute and too small near convergence to matter and so is omitted to give the so called Gauss Newton (GN) Hessian:

$$H_{gn} = \frac{\partial I^T}{\partial p} \frac{\partial^2 f}{\partial I^2} \frac{\partial I}{\partial p}$$  \hspace{1cm} (19)$$

Though $H_{gn}$ works very well for SSD (and in fact even better than $H$ [4,30]), it is well known [30,21] to not perform well with other AMs like MI, CCRE and NCC and we can confirm that the latter is true for SSIM and SPSS too. For these AMs, an approximation to the Hessian after convergence has to be used instead by assuming perfect alignment or $I_t(w(x_0, \hat{p}_t)) = I_0(x_0)$ where $\hat{p}_t$ is an estimate of $p_t$ (typically $p_{t-1}$) to which an incremental update is sought. We refer to the resultant approximation as the Self Hessian $H_{self}$ and, as this substitution can be made by setting either $I^c = I_0(x_0)$ or $I^* = I_t(w(x_0, \hat{p}_t))$, we get two forms which are respectively deemed to be the Hessians for ICLK and FCLK:

$$\hat{H}_{ic} = H^*_{self} = \frac{\partial I_0^T}{\partial p} \frac{\partial^2 f(I_0, I_0)}{\partial I^2} \frac{\partial I_0}{\partial p} + \frac{\partial f(I_0, I_0)}{\partial I} \frac{\partial^2 I_0}{\partial p^2}$$  \hspace{1cm} (20)$$

$$\hat{H}_{fc} = H^e_{self} = \frac{\partial I_t^T}{\partial p} \frac{\partial^2 f(I_t, I_t)}{\partial I^2} \frac{\partial I_t}{\partial p} + \frac{\partial f(I_t, I_t)}{\partial I} \frac{\partial^2 I_t}{\partial p^2}$$  \hspace{1cm} (21)$$

It is interesting to note that $H_{gn}$ has the exact same form as $H_{self}$ for SSD (since $\frac{\partial f_{ssd}(I_0, I_0)}{\partial I} = \frac{\partial f_{ssd}(I_t, I_t)}{\partial I} = 0$) so it seems that interpreting Eq. 19 as the first order approximation of Eq. 6 for SSD, as in [4,22,30], is incorrect and it should instead be seen as a special case of $H_{self}$.

Setting $I_0 = I_t$ makes $\mu_t = \mu_0$, $a' = c'$, $b' = d'$, $f = 1$ and $f' = 0$ so that Eq. 15 simplifies to:

$$\frac{\partial^2 f_{ssim}(I_t, I_t)}{\partial I_t^2} = -\frac{2}{c_{self} d_{self}} \left[ S_N \left( \frac{d_{self}}{N^2} 1_N \right) + \text{diag} \left( \frac{c_{self}}{N - 1} 1_N \right) \right]$$  \hspace{1cm} (22)$$

with $c_{self} = 2\mu_t^2 + C_1$ and $d_{self} = 2\sigma_t^2 + C_2$. Finally, FALK and IALK use the same form of $\frac{\partial^2 f}{\partial I^2}$ as FCLK while ESM uses the sum of $\hat{H}_{fc}$ and $\hat{H}_{ic}$.

### 3.2 Simplifying SSIM with pixelwise operations

In the formulation described so far, SSIM has been computed over the entire patch - i.e. $\mu_t$, $\sigma_t$ and $\sigma_{t0}$ have been computed over all $N$ pixels in $I_t$ and $I_0$. In its original formulation [10], however, the expression in Eq. 3 was applied to several corresponding sub windows within the two patches - for instance $8 \times 8$ or $11 \times 11$ sub windows that are moved pixel-by-pixel over the entire patch - and then the mean of all resultant scores was taken as the overall similarity score. Adopting such an approach is not only impracticable for tracking applications from the perspective of speed, it also presents another issue for gradient descent methods - presence of insufficient texture in these small sub windows may cause Eq. 4 to become ill posed if $J$ and $H$ are computed for each sub window and then averaged.
As a result, the formulation used here considers only one end of the spectrum of variation of size and number of sub windows - a single sub window of the same size as the patch. Now, if we consider the other end of the spectrum - \(N\) sub windows of size \(1 \times 1\) each - then we get a different AM that may give us some idea about the effect of window wise operations while also being much simpler and faster. We term the resultant model as **Sum of Pixelwise Structural Similarity** or SPSS. When considered pixel wise, \(\sigma_t\) and \(\sigma_{t0}\) become null while \(\mu_t\) becomes equal to the pixel value itself so that Eq. 3 simplifies to:

\[
f_{spss} = \sum_{i=1}^{N} \frac{2I_t(x_i^t)I_0(x_0^i) + C_1}{I_t(x_i^t)^2 + I_0(x_0^i)^2 + C_1}
\]  

(23)

Similar to SSD, the contributions from different pixels to \(f_{spss}\) are independent of each other so that each entry of the Jacobian has contribution only from the corresponding pixel. This also holds true for each entry of the principal diagonal of the Hessian (which is a diagonal matrix). Denoting the contributions of the \(i^{th}\) pixel to \(f_{spss}\), 

\[
\frac{\partial f_{spss}}{\partial I_t} \quad \text{and} \quad \frac{\partial^2 f_{spss}}{\partial I_t^2}
\]

respectively as \(f_i\), \(f_i'\) and \(f_i''\), we get:

\[
f_i' = \frac{2(I_0(x_0^i) - I_t(x_i^t)f_i)}{I_t(x_i^t)^2 + I_0(x_0^i)^2 + C_1}
\]  

(24)

\[
f_i'' = \frac{-2(f_i + 3I_t(x_i^t)f_i')}{I_t(x_i^t)^2 + I_0(x_0^i)^2 + C_1}
\]  

(25)

\[
f_i''(I_t, I_t) = \frac{-2}{2I_t(x_i^t)^2 + C_1}
\]  

(26)

4 Results and Analysis

4.1 Dataset and Error Metric

Three publicly available datasets have been used to analyze the trackers:

1. Tracking for Manipulation Tasks (TMT) dataset [31] that contains videos of some common tasks performed at several speeds and under varying lighting conditions. It has 109 sequences with a total of 70592 frames.

2. Visual Tracking Dataset provided by UCSB [32] that has 96 short sequences of different challenges in planar object tracking with a total of 6889 frames. The sequences here are more challenging but also rather artificial since they were created specifically to address various challenges rather than represent realistic scenarios.

3. LinTrack dataset [33] that has 3 long sequences with a total of 12477 frames. These are more realistic than those in UCSB but also more difficult to track.

All of these datasets have full pose (8 DOF) ground truth data which makes them suitable for evaluating high precision trackers that are the subject of this study. In addition, we use **Alignment Error** \((E_{AL})\) [18] as metric to compare tracking result with the ground truth since it accounts for fine misalignments of pose better than other common measures like center location error and Jaccard index.
4.2 Evaluation Measure

We measure a tracker’s overall accuracy through its success rate (SR) which is defined as the fraction of total frames where the tracking error $E_{AL}$ is less than a threshold of $t_p$ pixels. Formally, $SR = \frac{|S|}{|F|}$ where $S = \{f^i \in F : E_{AL}^i < t_p\}$, $F$ is the set of all frames and $E_{AL}^i$ is the error in the $i^{th}$ frame $f^i$. Since we have far too many sequences to present results for each, we instead report an overall summary of performance by averaging the success rates over all the sequences in the three datasets, i.e. $F$ is treated as the set of all frames in TMT, UCSB and LinTrack with $|F| = 70592 + 6889 + 12477 - 208 = 89750$ - we do not consider the first frame in each sequence, where the tracker is initialized, for computing the SR. Finally, we evaluate SR for several values of $t_p$ ranging from 0 to 20 and study the resulting SR vs. $t_p$ plot to get an overall idea of how precise and robust a tracker is. More detailed results, including those on individual datasets, are included in the supplementary.

4.3 Parameters Used

All results have been generated using a fixed sampling resolution of $50 \times 50$ irrespective of the tracked object’s size. The input images were smoothed using a Gaussian filter with a $5 \times 5$ kernel before being fed to the trackers. Iterative SMs were allowed to perform a maximum of 30 iterations per frame but only as long as the L2 norm of the change in bounding box corners in each iteration remained greater than 0.001. A standard deviation of 0.05 was used for generating the random warps for the NN tracker. For RANSAC and RKLT, a $5 \times 5$ grid of sub patches was used and each sub patch was tracked by a 2 DOF ICLK tracker with a sampling resolution of $25 \times 25$. As in [10], SSIM parameters are computed as $C_1 = (K_1L)^2$ and $C_2 = (K_2L)^2$ with $K_1 = 0.01$, $K_2 = 0.03$ and $L = 255$ The learning based trackers whose results are reported in Sec. 4.4 were run using default settings provided by their respective authors. All speed tests were performed on a 2.66 GHz Intel Core 2 Quad Q9450 machine with 4 GB of RAM. No multi threading was used.

4.4 Results

The results presented in this section are organized into three sections corresponding to the three sub modules. In each of these, we present and analyze results comparing different methods for the respective sub module with one or more combinations of methods for the other sub modules. SSM is fixed to homography for the first two sections.

**Search Methods** Fig. 1 presents the results for all SMs except NN and RANSAC which are given in Fig. 3(g,h). This exclusion was needed because these SMs, due to their stochastic nature, have significantly lower SR for smaller $t_p$ than other SMs. In order to maximize the visibility of individual curves in the plots within Fig. 1, the y axis in each has been limited to the range where the curves in that plot actually lie. Inclusion of these results here would have caused this range to increase significantly, decreasing the separation between the curves and making analysis more difficult. NN
Fig. 1: Success rates for SMs using different AMs with Homography. Note that the range of y axis varies between plots to maximize visibility. Best viewed on a high resolution screen.

Fig. 2: Success Rates for SSMs using different AMs with ESM. Note that homography has 3 parameterizations that overlap perfectly. These plots were generated using corresponding low DOF ground truth for each SSM.
Fig. 3: Success rates for AMs using different SMs with Homography. Best viewed on a high resolution screen. NN results are shown in dashed lines for NN1K and solid lines for NN10K.

Fig. 4: Success Rates for SSIM and SPSS using Translation as well as for 5 learning based trackers. The former are shown with solid lines and the latter in dashed lines. The speed plot on the right uses logarithmic scaling on the x axis for clarity though actual figures are also shown.
results are included for CCRE (Fig. 3(e)), however, as it performs poorly with all SMs so the inclusion does not matter.

First fact to observe from Fig. 1 is that the four variants of LK do not perform identically - FCLK is the best for all AMs and is significantly better than FALK especially for smaller $t_p$. ICLK and IALK, on the other hand, are more contentious, being very similar for 4 AMs - SPSS, SSD, RSCV and LSCV - but ICLK being appreciably better for the other five. This is especially true for CCRE where it is equivalent to ESM for larger $t_p$ and much better than both the additive variants. This finding contradicts the equivalence between these variants that was reported in \cite{4} and justified there using both theoretical analysis and experimental results. The latter, however, were only performed on synthetic images and even the former used several approximations. So, it is perhaps not surprising that this supposed equivalence does not hold under real world conditions.

Secondly, we note that ESM fails to outperform FCLK for any AM except MI and even there it does not lead by much. This fact too emerges in contradiction to the theoretical analysis in \cite{5} where ESM was shown to have second order convergence and so should be better than first order methods like FCLK. Thirdly, we see that both additive LK variants and especially IALK perform much poorer compared to the compositional variants with the robust AMs than with the SSD-like AMs. This is probably to be expected since the Hessian after convergence approach used for extending the GN method to these AMs does not make as much sense for additive formulations \cite{30}.

**Appearance Models** Fig. 3 shows the SR curves for all AMs except CCRE which is excluded for the same reason as NN/RANSAC in the previous section. This reason itself is the most obvious point to be noted - in spite of being the most sophisticated and computationally expensive AM, CCRE performs much poorer than other AMs with all SMs except those based on NN. In fact NN10K is the best performing SM for CCRE for all but very small $t_p$. Another interesting fact is that both CCRE and ZNCC actually perform far worse with NNIC than with either NN1K or ICLK which is very unexpected as the composite tracker uses inputs from both and so should perform at least as well as the best of these. We repeated these experiments several times but these discrepancies remained. Further, both MI and CCRE perform very poorly with RANSAC and RKLT, probably because the small sub patches used there do not have sufficient information to be represented well by the joint histograms used by these AMs.

The next fact to note is that NCC is the best performer with all SMs except IALK (which performs poorly with all robust AMs as noted in the previous section) though it is matched well by SSIM with all 4 of the stochastic SMs. Also, as expected, SSIM is much better than SPSS with all SMs, with the latter only managing to match the performance of SSD on average. Finally, though ZNCC is supposedly equivalent to NCC \cite{25} and also has a wider basin of convergence due to its SSD like formulation, it usually does not perform as well as NCC.

**State Space Models** The results presented in this section follow a slightly different format from the other two sections due to the difference in the motivations for using low DOF SSMs - they are more robust and faster. Limiting the DOF also makes regis-
tation based trackers directly comparable to learning based trackers as these too work in low DOF search spaces. As a result, we also present results for five state of the art learning based trackers [2] - discriminative scale space tracker (DSST), kernelized correlation filter tracker (KCF), tracking-learning-detection (TLD), real time compressive tracker (RCT) and consensus-based matching of keypoints tracker (CMT). Lastly, in order to make the evaluations fair, we have used lower DOF ground truths for all accuracy results in this section. These were generated for each SSM using least squares optimization to find the warp parameters that, when applied to the initial bounding box, will produce a warped box whose alignment error ($E_{AL}$) with respect to the full 8 DOF ground truth is as small as it is possible to achieve given the constraints of that SSM.

Fig. 4 shows the performance of all SMs with translation in terms of both accuracy and speed. As expected, all the learning based trackers have low SR for smaller $t_p$ since they are less precise in general [2]. What is more interesting, however, is that none of these trackers, with the exception of DSST, managed to surpass the best registration based trackers even for larger $t_p$ though they did close the gap. Even DSST only managed it at the extreme tail end of the plot and by a small margin. The superiority of DSST over other learning based trackers is at least consistent with results published elsewhere [2]. The speed comparisons in Fig. 4 clearly show the main reason why learning trackers are not suitable for high speed tracking scenarios - they are 10 to 30 times slower than their registration based counterparts.

Some interesting observations can be made by comparing the different SMs too. Firstly, FALK and FCLK show perfect overlap which is to be expected as the two formulations are identical for translation. Secondly, NN1K and NN10K have practically identical performance for both SSIM and SPSS even though NN10K does perform significantly better than NN1K with homography (Fig. 3(e)). It seems, however, that more samples do not help much with low DOF search spaces as 1000 samples is already enough to cover it well and it is the quality of samples that forms the bottleneck now. To conclude the analysis in this section, we tested the performance of different SSMs against each other and the results are reported in Fig. 2 using ESM with three AMs. Contrary to expectations, we find that lower DOF SSMs are not more robust except for larger $t_p$ where affine and similitude at least are better than homography. Also, all three parameterizations of homography have exactly identical performance with their plots showing perfect overlap. This indicates that the theoretical justification given in [5] for parameterizing ESM with $SL(3)$ has little practical significance.

5 Conclusions

We presented two new similarity measures for registration based tracking. We also formulated a novel method to decompose trackers in this domain into sub modules and tested several different combinations of methods for each sub module to gain interesting insights into the strengths and weaknesses of these methods. We also obtained some rather surprising results that proved previously published theoretical analysis to be somewhat inaccurate in practice, thus demonstrating the usefulness of our framework. We also make publicly available the open source tracking framework so that all results can be reproduced.
References

1. Wu, Y., Lim, J., Yang, M.H.: Online Object Tracking: A Benchmark. In: CVPR. (June 2013) 2411–2418
2. Kristan, Matej, M.J.L.A., et al.: The Visual Object Tracking VOT2015 Challenge Results. In: Proceedings of the IEEE ICCV Workshops. (2015) 1–23
3. Lucas, B.D., Kanade, T.: An Iterative Image Registration Technique with an Application to Stereo Vision. In: 7th international joint conference on Artificial intelligence. Volume 2., San Francisco, CA, USA, Morgan Kaufmann Publishers (1981) 674–679
4. Baker, S., Matthews, I.: Lucas-Kanade 20 Years On: A Unifying Framework. International Journal of Computer Vision 56(3) (Feb 2004) 221–255
5. Benhimane, S., Malis, E.: Homography-based 2D Visual Tracking and Servoing. Int. J. Rob. Res. 26(7) (July 2007) 661–676
6. Hager, G.D., Belhumeur, P.N.: Efficient Region Tracking With Parametric Models of Geometry and Illumination. IEEE Transactions on Pattern Analysis and Machine Intelligence 20(10) (October 1998) 1025–1039
7. Richa, R., Sznitman, R., Taylor, R., Hager, G.: Visual tracking using the sum of conditional variance. In: Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on. (Sept 2011) 2953–2958
8. Dame, A., Marchand, E.: Accurate real-time tracking using mutual information. In: IEEE International Symposium on Mixed and Augmented Reality. (2010) 47–56
9. Briechele, K., Hanebeck, U.D.: Template matching using fast normalized cross correlation. In: Aerospace/Defense Sensing, Simulation, and Controls, International Society for Optics and Photonics (2001) 95–102
10. Wang, Z., Bovik, A., Sheikh, H., Simoncelli, E.: Image quality assessment: from error visibility to structural similarity. Image Processing, IEEE Transactions on 13(4) (April 2004) 600–612
11. Łoza, A., Mihaylova, L., Bull, D., Canagarajah, N.: Structural similarity-based object tracking in multimodality surveillance videos. Machine Vision and Applications 20(2) (2009) 71–83
12. Wang, X., Ning, C., Shi, A., Lv, G.: An improved similarity measure in particle filters for robust object tracking. In: Image and Signal Processing (CISP), 2013 6th International Congress on. Volume 1., IEEE (2013) 46–50
13. Loza, A., Wang, F., Yang, J., Mihaylova, L.: Video object tracking with differential Structural SIMilarity index. In: Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on. (May 2011) 1405–1408
14. Loza, A., Wang, F., Patricio, M.A., García, J., Molina, J.M.: Optimised Particle Filter Approaches to Object Tracking in Video Sequences. In: Methods and Models in Artificial and Natural Computation. Springer (2009) 486–495
15. Rogerio Richa, Raphael Sznitman, G.H.: Robust Similarity Measures for Gradient-based Direct Visual Tracking. Technical report, CIRL (June 2012)
16. Singh, A., Jagersand, M.: Modular Tracking Framework: A Unified Approach to Registration based Tracking. submitted to CRV, available at:http://arxiv.org/abs/1602.09130 (2016)
17. Szeliski, R.: Image Alignment and Stitching: A Tutorial. Foundations and Trends in Computer Graphics and Vision 2(1) (January 2006) 1–104
18. Dick, T., Perez, C., Jagersand, M., Shademan, A.: Realtime Registration-Based Tracking via Approximate Nearest Neighbour Search. In: Proceedings of Robotics: Science and Systems, Berlin, Germany (June 2013)
19. Muja, M., Lowe, D.G.: Fast Approximate Nearest Neighbors with Automatic Algorithm Configuration. VISAPP (1) 2 (2009) 331–340
20. Zhang, X., Singh, A., Jagersand, M.: RKLT: 8 DOF real-time robust video tracking combining coarse RANSAC features and accurate fast template registration. In: CRV, IEEE (2015) 70–77

21. Scandaroli, G.G., Meilland, M., Richa, R.: Improving NCC-based Direct Visual Tracking. In: ECCV. Springer (2012) 442–455

22. Dowson, N., Bowden, R.: Mutual Information for Lucas-Kanade Tracking (MILK): An Inverse Compositional Formulation. PAMI 30(1) (Jan 2008) 180–185

23. Wang, F., Vemuri, B.C.: Non-rigid multi-modal image registration using cross-cumulative residual entropy. International journal of computer vision 74(2) (2007) 201–215

24. Richa, R., Souza, M., Scandaroli, G., Comunello, E., von Wangenheim, A.: Direct visual tracking under extreme illumination variations using the sum of conditional variance. In: Image Processing (ICIP), 2014 IEEE International Conference on. (Oct 2014) 373–377

25. Ruthotto, L.: Mass-preserving registration of medical images. German Diploma Thesis (Mathematics), Institute for Computational and Applied Mathematics, University of Münster (2010)

26. Jain, A., Nandakumar, K., Ross, A.: Score normalization in multimodal biometric systems. Pattern recognition 38(12) (2005) 2270–2285

27. Hartley, R., Zisserman, A.: Multiple View Geometry in Computer Vision. Second edn. Cambridge University Press (March 2004)

28. Bouguet, J.Y.: Pyramidal Implementation of the Affine Lucas Kanade Feature Tracker: Description of the algorithm. Technical report, Intel Corporation Microprocessor Research Labs (2001)

29. Brooks, R., Arbel, T.: Generalizing Inverse Compositional and ESM Image Alignment. International Journal of Computer Vision 87(3) (May 2010) 191–212

30. Dame, A.: A unified direct approach for visual servoing and visual tracking using mutual information. PhD thesis, University of Rennes (2010)

31. Roy, A., Zhang, X., Wolleb, N., Perez Quintero, C., Jagersand, M.: Tracking benchmark and evaluation for manipulation tasks. In: Robotics and Automation (ICRA), 2015 IEEE International Conference on, IEEE (2015) 2448–2453

32. Gauglitz, S., Höllerer, T., Turk, M.: Evaluation of interest point detectors and feature descriptors for visual tracking. International journal of computer vision 94(3) (2011) 335–360

33. Zimmermann, K., Matas, J., Svooba, T.: Tracking by an optimal sequence of linear predictors. Pattern Analysis and Machine Intelligence, IEEE Transactions on 31(4) (2009) 677–692