# Modular Decomposition and Analysis of Registration based Trackers

Abhineet Singh, Ankush Roy, Xi Zhang, Martin Jagersand
Department of Computing Science
University of Alberta, Edmonton, Canada
{asingh1, ankush2, xzhang6}@ualberta.ca, jag@cs.ualberta.ca

*Abstract*—**This paper presents a new way to study registration based trackers by decomposing them into three constituent sub modules: appearance model, state space model and search method. It is often the case that when a new tracker is introduced in literature, 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.**

**We show how existing trackers can be broken down using the suggested methodology and compare the performance of the default configuration chosen by the authors against other possible combinations to demonstrate the new insights that can be gained by such an approach. We also present an open source system that provides a convenient interface to plug in a new method for any sub module and test it against all possible combinations of methods for the other two sub modules while also serving as a fast and efficient solution for practical tracking requirements.**

## I. Introduction

Since its inception, research in object tracking has focused on presenting new tracking algorithms to address specific challenges in a wide variety of application domains like surveillance, targeting systems, augmented reality and medical analysis. Before an algorithm 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 [29], [16] 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 trackers 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 [18], [1] which - as we show in this work - can be decomposed into three well defined modules, thus making their systematic analysis feasible. These trackers are generally faster and more precise than learning based trackers [23] which makes them more suitable for applications such as robotic manipulations, visual servoing and SLAM, where multiple trackers are used in parallel. On the other hand, lacking an online learning component, they are known to be non robust to changes in the object's appearance and prone to failure in the presence of motion blur, occlusion, lighting variations or viewpoint changes. As a result, they are less popular in the vision community and often underrepresented in the aforementioned studies, thus making such an evaluation particularly useful for applications where learning based trackers are unsuitable. A detailed analysis, with a test framework in registration based tracking, to the best of our knowledge. has never been attempted before.

Many reported studies in this domain [18], [1], [2] have introduced new methods for only one of the three submodules without exploring the full extent of their contributions. For instance, Baker et. al [1] reported a compositional update scheme for the state parameters $\mathbf{p}$ (Eq. 1) instead of the additive scheme used in [12], but did not experiment with different similarity metrics. Conversely, Richa et. al [21] showed an improvement over the existing efficient second order minimization [2] approach by using the sum of conditional variance as the similarity metric instead of the sum of squared differences. Similarly, Dame et. al [7] used mutual information while Scandaroli et. al [4] used normalized cross correlation with the inverse compositional method of [1]. However, neither of them tested their similarity measures with other search methods even though the latter had previously been shown to be a good metric when used with the standard Lucas Kanade type tracker [4].

Finding the optimal combination of methods for any tracking algorithm is a two step process. First, the sub module where the algorithm's main contribution lies needs to be determined, using, for instance, the method employed in [29]. Second, all possible combinations for the other sub modules that are compatible with this algorithm (since not all methods for different sub modules work with each other) need to be enumerated and evaluated. A generic framework would thus be useful to avoid such fragmentation.

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

- Empirically test different combinations of submodules leading to several interesting observations and insights that were missing in the original papers. Experiments are done using two large datasets with over 77,000 frames in all to ensure their statistical significance.

- Report for the first time, to the best of our knowledge, results comparing robust similarity metrics [22], with traditional SSD type measures.
- Compare formulations against popular online learning based trackers to validate their usability in precise tracking applications.
- Provide an open source tracking framework [1] using which all results can be reproduced and which, owing to its efficient C++ implementation, can also be used to address practical tracking requirements.

## II. Descriptions of submodules

A registration based tracker can be decomposed into three sub modules: appearance model (**AM**), state space model (**SSM**) and search method (**SM**). Figure 1 shows how these modules work together in a complete tracking system.

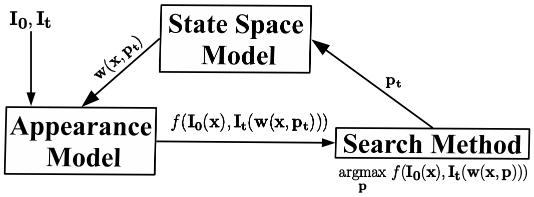
$\mathbf{I_0}, \mathbf{I_t}$
State Space Model
$\mathbf{w(x, p_t)}$
$\mathbf{p_t}$
Appearance Model
$f(\mathbf{I_0(x)}, \mathbf{I_t(w(x, p_t))})$
Search Method
$\underset{\mathbf{p}}{\text{argmax}} \, f(\mathbf{I_0(x)}, \mathbf{I_t(w(x, p))})$

Fig. 1: Modular breakdown of a registration based tracker assuming there is no dynamic update to the object template.

When a geometric transform $\mathbf{w}$ with parameters $\mathbf{p} = (p_1, p_2, ..., p_S)$ is applied to an image patch $\mathbf{x}$, the transformed patch is denoted by $\mathbf{x}' = \mathbf{w}(\mathbf{x}, \mathbf{p})$ and the corresponding pixel values in image $I$ as $\mathbf{I}(\mathbf{w}(\mathbf{x}, \mathbf{p}))$. Tracking can then be formulated (Eq 1) as a search problem where we need to find the optimal transform parameters $\mathbf{p_t}$ for an image $I_t$ that maximize the similarity, measured by a suitable metric $f$, between the target patch $\mathbf{I}^* = \mathbf{I_0}(\mathbf{w}(\mathbf{x}, \mathbf{p_0}))$ and the warped image patch $\mathbf{I_t}(\mathbf{w}(\mathbf{x}, \mathbf{p_t}))$.

$$\mathbf{p_t} = \underset{\mathbf{p}}{\text{argmax}} \; f(\mathbf{I}^*, \mathbf{I_t}(\mathbf{w}(\mathbf{x}, \mathbf{p}))) \tag{1}$$

We refer to the similarity metric $f$, the warp function $\mathbf{w}$ and the algorithm that maximizes Eq 1 respectively as AM, SSM and SM. A more detailed description of these submodules follows.

### A. Search Method

This is the optimization procedure that searches for the warped patch in the current image that best matches the original template. Gradient descent is the most popular optimization approach used in tracking due to its speed and simplicity and is the basis of the classic Lucas Kanade (LK) tracker [18]. This algorithm can be formulated in four different ways [1] depending on which image is searched for the warped patch - $I_t$ or $I_0$ - and how the parameters of the warping function are updated in each iteration - additive or compositional. The four resulting variants - forward additive (**FALK**) [18], inverse additive (**IALK**) [12], forward compositional (**FCLK**) [27] and inverse compositional (**ICLK**) [1] - were analyzed mathematically and shown to be equivalent to first order terms in [1]. Here, however, we show experimental results proving that their performance on real video benchmarks is quite different (Sec. III-D1).

A relatively recent update to this approach was in the form of the Efficient Second order Minimization (**ESM**) [2] technique that tries to make the best of both inverse and forward formulations by using the mean of the initial and current Jacobians. We would like to mention here that, even though the authors of [2] used $\mathbb{SL}(3)$ parameterization for their ESM formulation and gave theoretical proofs as to why it is essential for this SM, we have used standard parameterization (i.e. using matrix entries [27], [1]) for all our experiments since, as we show later (Sec. III-D3), ESM actually performs identically with several different parameterizations.

Further, since the standard formulations for these SMs using the Gauss Newton Hessian [18], [1], [2] do not work with any AMs besides SSD [7], [25], a modified version with the so called *Hessian after convergence* [7], [25] has been used for our experiments. Also, the extended formulation for ESM reported in [5], [25] has been used instead of the original one in [2]. The exact formulations used can be found in [26].

Nearest neighbor search (NN) is another SM that has recently been used for tracking [8] 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, as this method tends to give jittery and unstable results when used by itself due to the very limited search space, it was used in conjunction with a gradient descent type SM in [8] to create a composite tracker that performs better than either of its constituents. As in [8], we have used ICLK as this second tracker due to its speed and the resultant composite SM is named **NNIC**. Unlike NN, results for NNIC are only reported using 1000 samples for NN as NN10K is too slow to be combined with ICLK.

### B. 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**) [18], [1], [2] or the L2 norm of pixel differences is the AM used most often 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 lighting variations, motion blur and occlusion.

To address these issues, more robust AMs have been proposed including Sum of Conditional Variance (**SCV**) [21],

[1] available online at http://webdocs.cs.ualberta.ca/~vis/mtf/

Normalized Cross Correlation (**NCC**) [25], Mutual Information (**MI**) [9], [7] and Cross Cumulative Residual Entropy (**CCRE**) [28], [22], 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**) [8] where $\mathbf{I_t}$ is updated rather than $\mathbf{I_0}$. There has also been a recent extension to it called **LSCV** [20] that uses multiple joint histograms from corresponding sub regions within the target patch to achieve greater robustness to localized intensity changes. It has further been shown [24] that maximizing NCC between two images is equivalent to minimizing the SSD between two z-score [15] normalized images. We consider the resultant formulation as a different AM called Zero Mean NCC (**ZNCC**).

It may ne noted that these AMs can be divided into 2 distinct categories - those that use some form of the L2 norm as the similarity function - SSD, SCV, RSCV, LSCV and ZNCC - and those that do not - MI, CCRE and NCC. The latter are henceforth called robust models after [22].

### *C. State Space Model*

The SSM 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 [2], can be considered to have a different SSM than conventional LK type trackers [18], [1] even though both involve 8 DOF homography, since it uses the $\mathbb{SL}(3)$ parameterization rather than the actual entries of the corresponding matrix. We model 7 different SSMs - translation, isometry, similitude, affine and homography [27] along with two extra parameterizations of homography - $\mathbb{SL}(3)$ 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 [13] 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 [3] whose higher DOF variants are usually less robust.

It may be noted that this sub module differs from the other two in that it does not admit new methods in the conventional sense and may even be viewed as a part of the SM with the two being often closely intertwined in practical implementations. However, though the SSMs used in this work are limited to the standard hierarchy of geometric transformations, more complex models like piecewise projective transforms do exist and it is also theoretically possible to impose novel constraints on the search space that can significantly decrease the search time while still producing sufficiently accurate results. The fact that such a constraint will be an important contribution in its own right justifies the use of SSM as a sub module in this work to motivate further research in this direction.

## III. Experimental Results and Analysis

### *A. Dataset and Error Metric*

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

1) Tracking for Manipulation Tasks (**TMT**) dataset [23] 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** [11] that has 96 short sequences of different challenges in 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.

Both 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** ($E_{AL}$) [8] as the 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.

### *B. 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 both datasets, i.e. $F$ is treated as the set of all frames in TMT and UCSB with $|F| = 70592 + 6889 - 205 = 77276$ - 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.

### *C. Parameters Used*

All results have been generated using a fixed sampling resolution of 50×50 irrespective of the tracked object's size. The input images were smoothed using a Gaussian filter with a 5×5 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. For the NN tracker, a standard deviation of 0.05 was used for generating the random warps. The learning based trackers whose results are reported in Sec. III-D3 were run using default settings provided by their respective authors. All speed tests were run on a 2.66 GHz Intel Core 2 Quad Q9450 machine with 4 GB of RAM. No multi threading was used.

### *D. 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 implementing 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.

*1) Search Methods:* Fig. 3 presents the results for all SMs except NN1K and NN10K which are presented separately in Fig. 5. This separation was needed because NN, due to its stochastic nature, tends to have significantly lower SR for smaller thresholds than other SMs. In order to maximize the visibility of individual curves in the various plots within Fig. 3, the y axis in each has been limited to the range where the curves in that plot actually lie. Inclusion of NN results here would have caused this range to increase significantly, thus decreasing the separation between these curves and making analysis more difficult. SCV and CCRE results are excluded here too, the former because they are very similar to LSCV while the latter are presented separately in Fig. 2 for the same reason as NN but now pertaining to Fig. 4. Several interesting observations can be made from Figs. 3 and 2. Firstly, we see 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 thresholds. ICLK with IALK, on the other hand, are more contentious, being very similar for three AMs - SSD, RSCV and LSCV - but ICLK being appreciably better for the other four. This is especially true for CCRE where it is almost equivalent to FCLK for larger $t_p$ and much better than both the additive variants. This finding contradicts the equivalence between these variants that was reported in [1] 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.

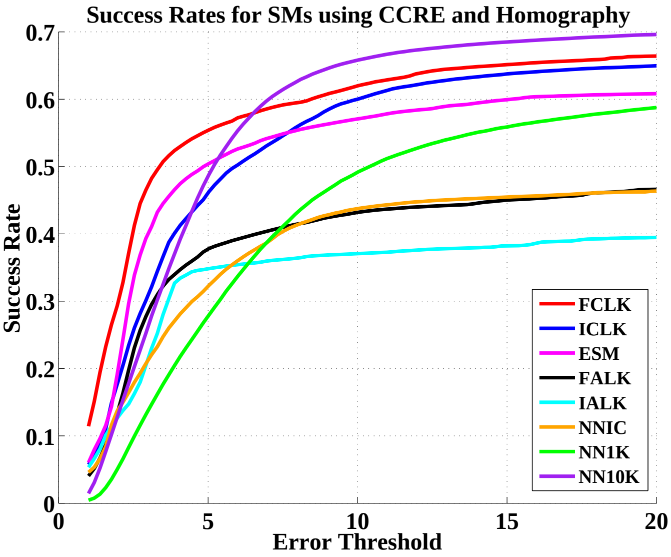

Fig. 2: Success rates for SMs using CCRE with Homography

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 [2] where ESM was shown to have second order convergence and so should be better than first order methods including FCLK. It might be argued that the extended version of ESM [5], [25] used here might not possess the characteristics of the formulation described in [2] but we conducted extensive experiments with that exact formulation too and can confirm that the version reported here performs identically to that one.

Thirdly, we see that NNIC does not perform better than ICLK on any of the AMs and is in fact significantly poorer with ZNCC. This yet again does not agree with the results reported in [8] using both static experiments and the Metaio dataset [17]. We have already seen in our first observation that static experiments may not always agree with real world tests and it must be admitted that sequences in the Metaio benchmark are highly artificial in nature as they neither represent real tasks nor include an actual background around the tracked patch. We did try to perform experiments on this dataset to check for possible bugs in our implementation but unfortunately the Metaio evaluation service is no longer available. However, to the best of our belief, there is no such bug and the discrepancy does indeed arise from the differences between artificial and real world benchmarks.

Fourthly, we can note 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 Gauss Newton method to these AMs does not make as much sense for additive formulations [6].

We conclude this section by examining the effect of number of samples on NN as well as its relative performance to gradient descent SMs from Figs. 2 and 5. We can see by comparing these plots to Fig. 4 that NN performs better relative to the latter with the robust AMs and in fact CCRE actually fares best with NN10K for larger $t_p$. This might indicate that the poor performance of CCRE, and to an extent MI, with LK type trackers has more to do with gradient descent optimization itself rather than some limitation of these AMs as good similarity metrics. The gain in performance between NN10K and NN1K though seems to be similar for all AMs as it is caused by an improved coverage of the SSM search space and so should depend only on that.

*2) Appearance Models:* Fig. 4 shows the SR curves for all AMs except CCRE whose results are in Fig. 2 for reasons

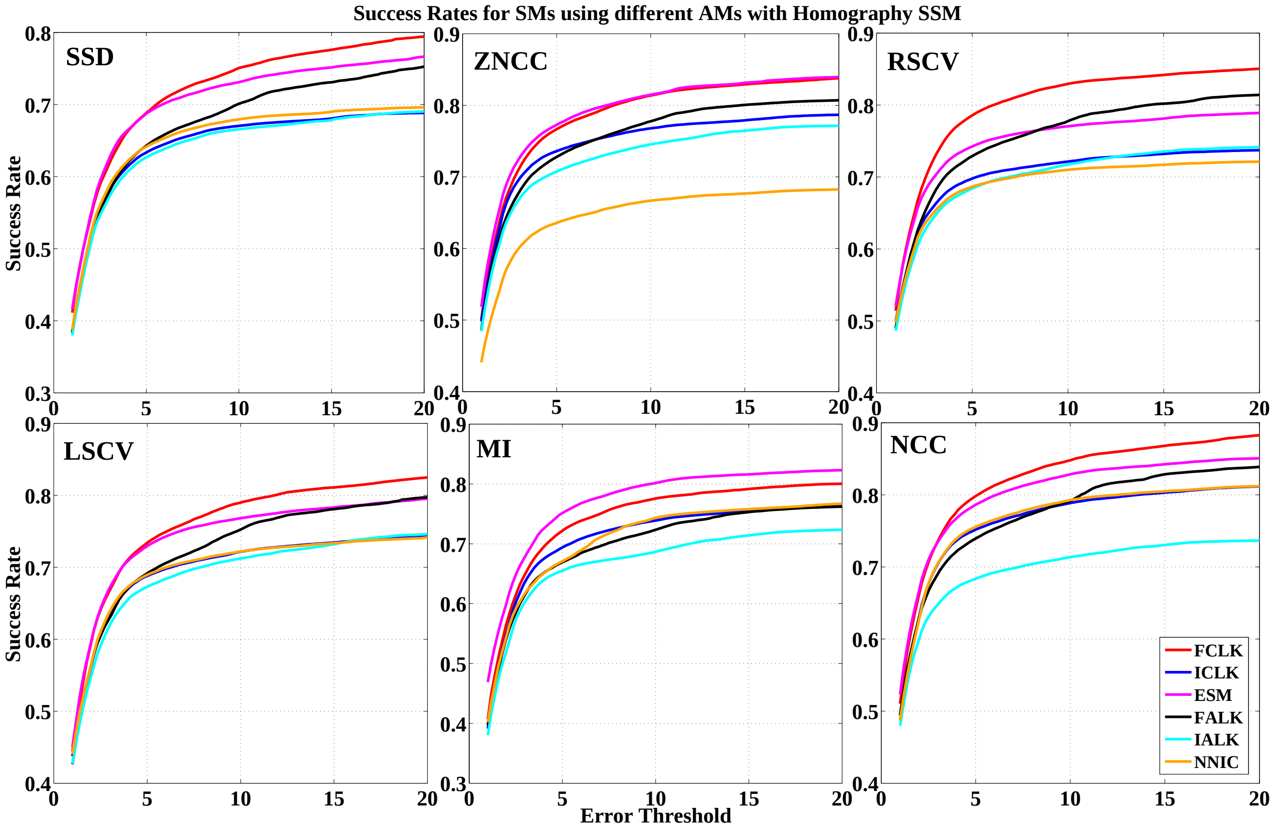

Fig. 3: Success rates for SMs using Homography SSM and different AMs. Best viewed on a high resolution screen.

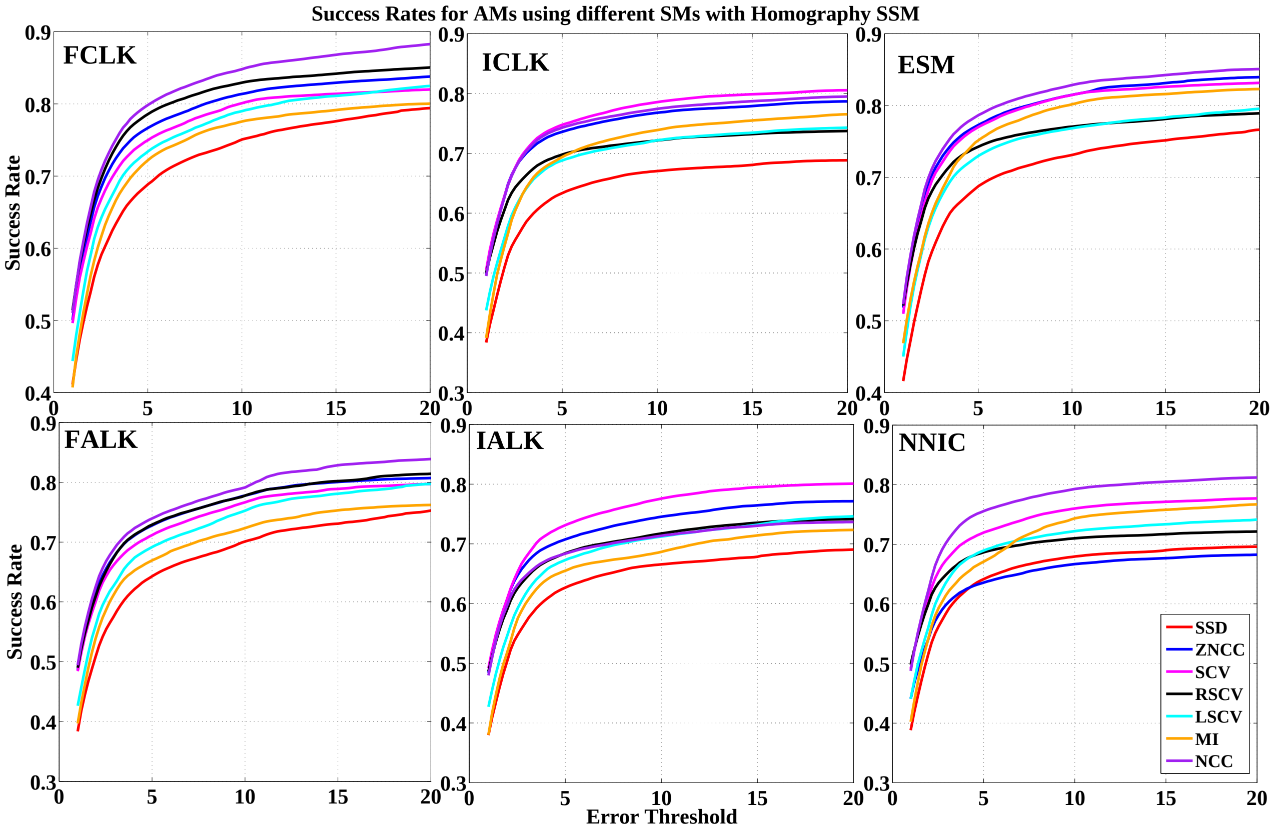

Fig. 4: Success rates for AMs using Homography SSM and different SMs. Best viewed on a high resolution screen.

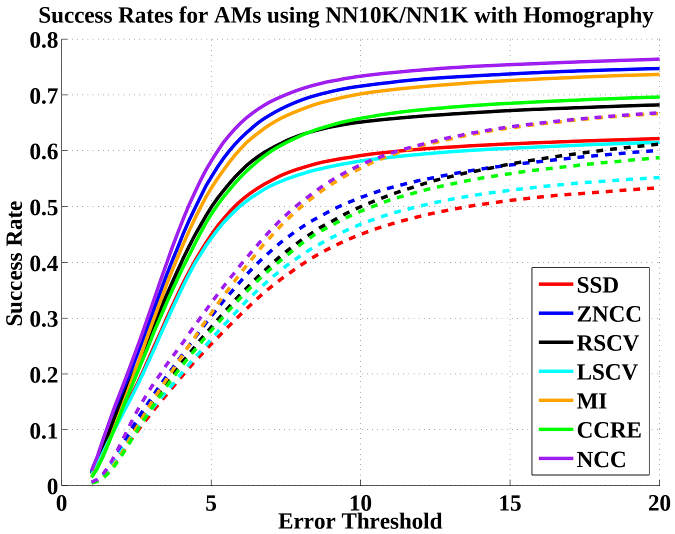

Fig. 5: Success rates for AMs using NN10K and NN1K with Homography represented with **solid and dashed lines** respectively. SCV, being almost identical to LSCV, has been ommitted for clarity.

already mentioned in the previous section. This reason itself is the most obvious point to be noted by comparing Figs. 4 and 2 - that CCRE, even though it is the most sophisticated and computationally expensive AM, performs much poorer than other AMs with all SMs except those based on NN. Another interesting fact is that it actually performs far worse with NNIC than it does 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. A similar phenomenon can be observed with ZNCC too. We repeated these experiments several times but these discrepancies remained.

Further, even MI is only slightly better than SSD on average, except with NN where it is among the best, being almost at par with NCC. It is much better than CCRE, however, in spite of the two AMs differing only in the latter using a cumulative joint histogram. It seems likely that the additional complexity of CCRE along with the resultant invariance to appearance changes significantly *reduces* its basin of convergence [7]. This leads to poor performance with gradient descent type SMs but, as expected, does not affect the efficacy of stochastic SMs.

The next fact to note is that NCC is the best performer with all SMs except IALK (which performs poorly with all robust AMs anyway as noted in the previous section). We also note that, though ZNCC is supposedly equivalent to NCC [24] and also has a wider basin of convergence due to its SSD like formulation, it usually does *not* perform as well as NCC. However both ZNCC and NCC are almost always better than SCV and its extensions LSCV/RSCV.

This last observation is rather contrary to expectations since SCV is supposedly more robust against lighting changes due to its use of joint probability distributions while ZNCC is merely the L2 norm between the pixel values normalized to have zero mean and unit variance. We can note too that LSCV, notwithstanding, its reported [20] increased invariance to localized intensity changes, fails to offer any improvement over either SCV or RSCV even though several of the tested sequences do exhibit such lighting changes. Considering that SCV and its variants are significantly more expensive than ZNCC to compute, there seems little reason to use these instead as the computational savings from ZNCC can be used to employ other ways (i.e. higher sampling resolution or more iterations) to improve performance.

*3) State Space Models:* The results presented in this section follow a slightly different format from the last two sections due to the difference in the motivations for using low DOF SSMs - the principle one being that reducing the dimensionality of the search space of warp parameters decreases the likelihood of the search process getting stuck in a local optimum, thus making the tracker more robust. The other less important motivation is that lower DOF SSMs tend to be faster since their Jacobians are typically less expensive to compute.

Limiting the DOF also makes registration based trackers directly comparable to learning based trackers as these too work in low DOF search spaces. As a result, in this section, we also present results for five state of the art learning based trackers [16] - 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**). We have used C++ implementations of all these trackers that are fully integrated into our framework. This not only makes it easy to reproduce the results presented here and but also makes it reasonable to compare the speeds of these trackers with the faster registration based trackers since slower speed is one of the main reasons why learning trackers are often not used in robotics applications.

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. In most cases, the ground truth corners thus generated represent the best possible performance that can theoretically be achieved by any tracker that follows the constraints of that SSM. In some rare cases, however, the resulting corners can be quite unexpected so we also visually inspected all lower DOF corners and corrected any that appeared unreasonable.

Fig. 6 shows the performance of all SMs with translation SSM in terms of both accuracy, evaluated against 2 DOF ground truth, and speed, measured in terms of the average number of frames processed by the tracker per second (FPS). In addition to the SMs described in Sec. II-A, results from another SM based on particle filter [14], generated using 1000 particles (**PF1K**), are also reported here. This is another

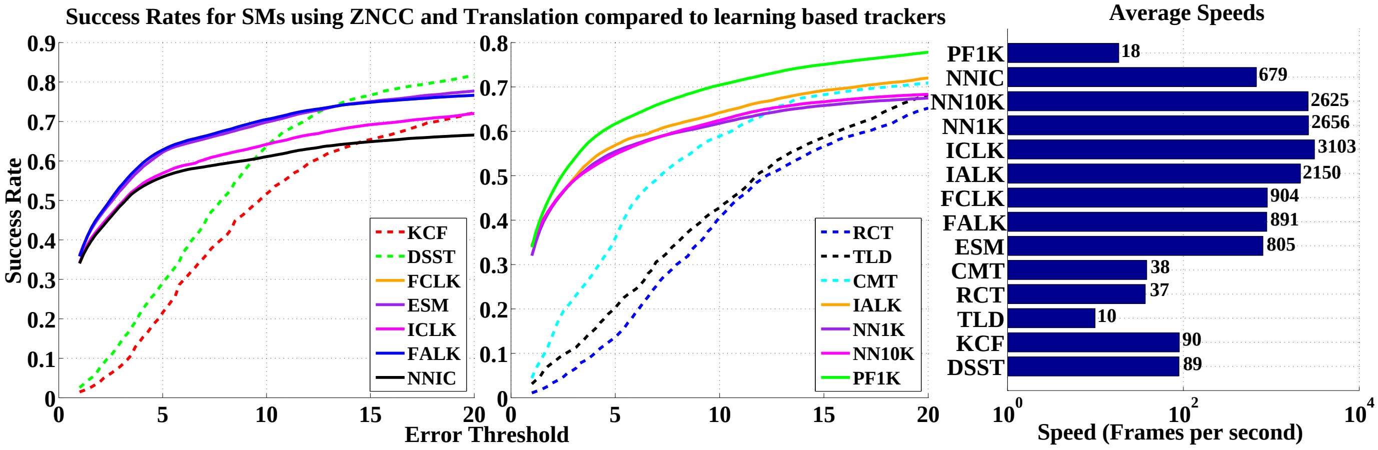

Fig. 6: Success Rates for SMs using ZNCC and Translation as well as for 5 learning based trackers. The former are shown with **solid** lines and the latter in **dashed** lines. 2 DOF ground truth was used for all evaluations. Note that the speed plot on the right uses **logarithmic scaling** on the x axis to increase visibility of the latter though the actual figures are mentioned too.

stochastic SM like NN that, though present in our framework, only works well with translation at the time of this writing and is thus not mentioned in the previous sections.

As expected, all the learning based trackers have low SR for smaller $t_p$ since they are less precise in general [16]. 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 [16].

The speed comparisons in Fig. 6 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. It may be noted that the speeds of the former depend on the size of the initial bounding box and so varied widely between sequences unlike the latter where a fixed sampling resolution was used. However, the mean figures reported here do provide a good idea of the general performance that can be expected from these trackers. It is not surprising that tracking based SLAM systems like SVO [10] use registration based trackers as they may need to track hundreds to thousands of patches per frame.

Some interesting observations can be made by comparing the different SMs too. Firstly, we see that FALK and FCLK show perfect overlap which is to be expected as the two formulations are identical for translation. Secondly, we note that NN1K and NN10K have practically identical performance in terms of both accuracy and speed. The latter is due to the KD Tree index used by FLANN library [19] being largely independent of the number of samples - only the initialization time increases when a larger index is to be built. The former is a bit more difficult to explain since NN10K does perform significantly better than NN1K with homography (Fig. 5). 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. This may be improved, for instance, by using different standard deviations to generate samples for each sequence which has not been done here. It may be noted too that PF performs at par with the best registration trackers. This is unsurprising since PF is known to perform well with low DOF when large number of particles are available - an advantage that comes at the cost of much slower speed.

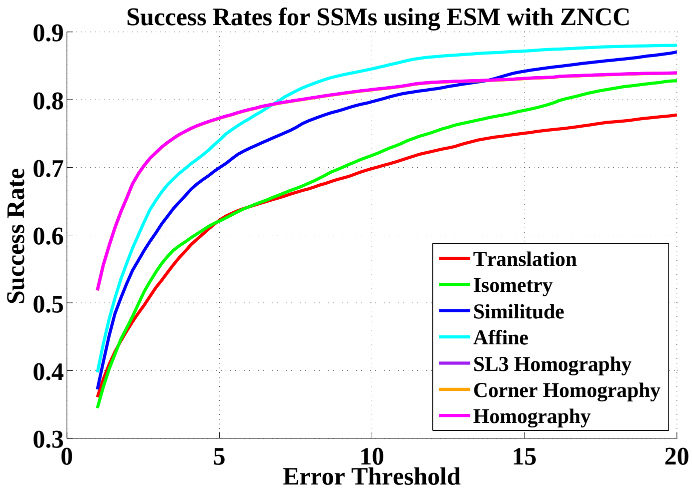

Fig. 7: Success Rates for all SSMs using ESM with ZNCC. Note that homography has 3 parameterizations that overlap perfectly. These plots were generated using corresponding low DOF ground truth for each SSM.

To conclude the analysis in this section, we tested the performance of different SSMs against each other and the results are reported in Fig. 7 using ESM with ZNCC. The plots for each SSM were generated by using the corresponding low DOF ground truth. As stated before, we were expecting lower

DOF SSMs to perform better here but this is not actually the case as higher DOF variants seem to perform better with the exception of affine which outperforms homography for larger values of $t_p$. However, the increased robustness of low DOF SSMs is at least partially apparent in the fact that their curves approach those of homography as $t_p$ increases with several surpassing it too. Thus, though they may not be as precise as homography, they do tend to be more resistant to complete drift. In fact, a general trend noticeable from the SR plots for high DOF SSMs, not only in Fig. 7 but also others analyzed earlier, is that, unlike low DOF SSMs (and learning based trackers), their SR does not continue to increase through the entire range of $t_p$ but instead flattens out after a certain point (often for $t_p < 10$). This results from the fact that, as long as these trackers work, they track the object very precisely but once they diverge, they do not drift off gradually but rather lose track quite abruptly.

Finally, it can be noted that all three parameterizations of homography have exactly identical performance with their plots showing perfect overlap. This indicates that the theoretical justification given in [2] for parameterizing ESM with $\mathbb{SL}(3)$ has little practical significance. This, in turn, may also suggest that, contrary to the assumption in [2], the reason for ESM's superior performance has more to do with its use of the information from both $I_0$ and $I_t$ rather than with it providing a pseudo second order convergence (opposed to LK's first order convergence).

## IV. Conclusions and Future Work

We formulated a novel method to decompose registration based trackers 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 in testing out new ideas in the domain of registration based tracking. We also make publicly available the open source modular tracking framework so all results can be reproduced. This framework, with its highly efficient and ROS compatible C++ implementations for several well established trackers, will hopefully address practical tracking needs of the wider robotics community too.

## References

[1] S. Baker and I. Matthews. Lucas-Kanade 20 Years On: A Unifying Framework. *International Journal of Computer Vision*, 56(3):221–255, Feb 2004. 1, 2, 3, 4

[2] S. Benhimane and E. Malis. Homography-based 2D Visual Tracking and Servoing. *Int. J. Rob. Res.*, 26(7):661–676, July 2007. 1, 2, 3, 4, 8

[3] J.-Y. Bouguet. Pyramidal Implementation of the Affine Lucas Kanade Feature Tracker: Description of the algorithm. Technical report, Intel Corporation Microprocessor Research Labs, 2001. 3

[4] K. Briechle and U. D. Hanebeck. Template matching using fast normalized cross correlation. In *Aerospace/Defense Sensing, Simulation, and Controls*, pages 95–102. International Society for Optics and Photonics, 2001. 1

[5] R. Brooks and T. Arbel. Generalizing Inverse Compositional and ESM Image Alignment. *International Journal of Computer Vision*, 87(3):191–212, May 2010. 2, 4

[6] A. Dame. *A unified direct approach for visual servoing and visual tracking using mutual information*. PhD thesis, University of Rennes, 2010. 4

[7] A. Dame and E. Marchand. Accurate real-time tracking using mutual information. In *IEEE International Symposium on Mixed and Augmented Reality*, pages 47–56, 2010. 1, 2, 3, 6

[8] T. Dick, C. Perez, M. Jagersand, and A. Shademan. Realtime Registration-Based Tracking via Approximate Nearest Neighbour Search. In *Proceedings of Robotics: Science and Systems*, Berlin, Germany, June 2013. 2, 3, 4

[9] N. Dowson and R. Bowden. Mutual Information for Lucas-Kanade Tracking (MILK): An Inverse Compositional Formulation. *PAMI*, 30(1):180–185, Jan 2008. 3

[10] C. Forster, M. Pizzoli, and D. Scaramuzza. SVO: Fast semi-direct monocular visual odometry. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pages 15–22. IEEE, 2014. 7

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

[12] G. D. Hager and P. N. Belhumeur. Efficient Region Tracking With Parametric Models of Geometry and Illumination. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(10):1025–1039, October 1998. 1, 2

[13] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, second edition, March 2004. 3

[14] M. Isard and A. Blake. CONDENSATION - conditional density propagation for visual tracking. *International Journal of Computer Vision*, 29:5–28, 1998. 6

[15] A. Jain, K. Nandakumar, and A. Ross. Score normalization in multimodal biometric systems. *Pattern recognition*, 38(12):2270–2285, 2005. 3

[16] M. J. L. A. Kristan, Matej et al. The Visual Object Tracking VOT2015 Challenge Results. In *Proceedings of the IEEE ICCV Workshops*, pages 1–23, 2015. 1, 6, 7

[17] S. Lieberknecht, S. Benhimane, P. Meier, and N. Navab. A dataset and evaluation methodology for template-based tracking algorithms. In *SMAR.*, pages 145–151. IEEE, 2009. 4

[18] B. D. Lucas and T. Kanade. An Iterative Image Registration Technique with an Application to Stereo Vision. In *7th international joint conference on Artificial intelligence*, volume 2, pages 674–679, San Francisco, CA, USA, 1981. Morgan Kaufmann Publishers. 1, 2, 3

[19] M. Muja and D. G. Lowe. Fast Approximate Nearest Neighbors with Automatic Algorithm Configuration. *VISAPP (1)*, 2:331–340, 2009. 2, 7

[20] R. Richa, M. Souza, G. Scandaroli, E. Comunello, and A. von Wangenheim. Direct visual tracking under extreme illumination variations using the sum of conditional variance. In *Image Processing (ICIP), 2014 IEEE International Conference on*, pages 373–377, Oct 2014. 3, 6

[21] R. Richa, R. Sznitman, R. Taylor, and G. Hager. Visual tracking using the sum of conditional variance. In *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, pages 2953–2958, Sept 2011. 1, 2

[22] G. H. Rogerio Richa, Raphael Sznitman. Robust Similarity Measures for Gradient-based Direct Visual Tracking. Technical report, CIRL, June 2012. 2, 3

[23] A. Roy, X. Zhang, N. Wolleb, C. Perez Quintero, and M. Jagersand. Tracking benchmark and evaluation for manipulation tasks. In *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, pages 2448–2453. IEEE, 2015. 1, 3

[24] L. Ruthotto. Mass-preserving registration of medical images. *German Diploma Thesis (Mathematics), Institute for Computational and Applied Mathematics, University of Münster*, 2010. 3, 6

[25] G. G. Scandaroli, M. Meilland, and R. Richa. Improving NCC-based Direct Visual Tracking. In *ECCV*, pages 442–455. Springer, 2012. 2, 4

[26] A. Singh and M. Jagersand. Modular Tracking Framework: A Unified Approach to Registration based Tracking. submitted to CRV, available at:http://arxiv.org/abs/1602.09130, 2016. 2

[27] R. Szeliski. Image Alignment and Stitching: A Tutorial. *Foundations and Trends in Computer Graphics and Vision*, 2(1):1–104, January 2006. 2, 3

[28] F. Wang and B. C. Vemuri. Non-rigid multi-modal image registration using cross-cumulative residual entropy. *International journal of computer vision*, 74(2):201–215, 2007. 3

[29] Y. Wu, J. Lim, and M.-H. Yang. Online Object Tracking: A Benchmark. In *CVPR*, pages 2411–2418, June 2013. 1