TST/BTD: An End-to-End Visual Recognition System

Taehee Lee Stefano Soatto

Technical Report UCLA-CSD100008

February 8, 2010, Revised March 18, 2010

Abstract

We describe a visual recognition system operating on a hand-held device. Feature selection and tracking are performed in real-time, and used to train a template-based classifier during a capture phase prompted by the user. During normal operation, the system scores objects in the field of view based on their ranking. Severe resource constraints have prompted a re-evaluation of existing algorithms improving their performance (accuracy and robustness) as well as computational efficiency. We motivate the design choices in the implementation with a characterization of the stability properties of local invariant detectors, and of the conditions under which a template-based descriptor is optimal. The analysis also highlights the role of time as "weak supervisor" during training, which we exploit in our implementation.

1 Introduction

In an attempt to implement a visual recognition system on a hand-held device, we chose off-the-shelf algorithms, suitably simplified to fit into the tight computational constraints. The disappointing overall performance prompted a re-evaluation of the algorithm from successive simplifications of basic models and assumptions. This has guided the architecture of the algorithm and its separate modules, including feature selection, tracking and description. The resulting modules are reminiscent of existing algorithms, but different in ways that yield better performance while reducing their computational cost. We describe our implementation in sect. 3, and the analysis that motivates the design choices in sect. 2.

1.1 Summary of contributions and relation to prior work

Our effort relates to a wealth of recent work on visual recognition that cannot realistically be reviewed here. We refer the reader to the PASCAL challenge [1] for references and comparisons of existing approaches. Our effort to run in real-time relates to [2], but resource constraints do not allow us to use sophisticated classification schemes such as random forests. Instead, we choose to work with simple classifiers (nearest neighbors and TF-IDF [3]) and focus on *representation* as the core issue. Modules of our system relate to multi-scale feature selection, tracking, local descriptors, and bag-of-features classification, specifically on *baseline algorithms* [4, 5, 6, 7] that we first intended to *"dumb-down"* to fit a hand-held platform, but ended up *improving* instead. We propose *a method to integrate multi-scale detection and tracking* that does *not* involve joint location-scale optimization [8], but explicitly accounts for topological changes across scales. This approach (dubbed "tracking on the selection tree", TST) respects the semi-group stucture of scaling/quantization, and is motivated by the "structural stability" of the selection process. This improves accuracy and robustness while making tracking more efficient. We also replace traditional single-view descriptors [6, 9, 10] with a *template* that is designed to be optimal in the mean-square sense, under conditions described in sect. 2, dubbed "best template" descriptor (BTD). Our contributions in this manuscript are the TST (sect. 2.4), the BTD (sect. 2.6), their motivation and analysis (sect. 2), and an iPhone implementation (sect. 3).

2 Representation

This section motivates our algorithm design choices via analysis of an abstraction of the recognition problem. The reader interested in just the algorithmic aspect of the system can skip ahead to sect. 3.

A grayscale image $I : \Omega \subset \mathbb{R}^2 \to \mathbb{R}_+;\ x \mapsto I(x)$ is generated by a scene $\xi = \{S, \rho\} \in \Xi$ of piecewise smooth surfaces $S \subset \mathbb{R}^3$ and albedo $\rho : S \to \mathbb{R}^k$. *Nuisances* $\{g, \nu\}$ are divided into those that are a group $g \in G$ (contrast transformations, local changes of viewpoint) and a non-invertible map ν (quantization, occlusions). Deviations from this model (non-diffuse reflectance, mutual illumination, cast shadows, sensor noise) are not represented explicitly and lumped as an additive error n:

$$I = h(g\xi, \nu) + n. \tag{1}$$

As abstract "visual recognition" tasks we consider *classifications* (detection, localization, categorization and recognition) that boil down to learning and evaluating the *likelihood* $p(I|c)$ of a class c that affects the data via a Markov chain $c \to \xi \to I$. For simplicity we only consider binary symmetric 0-1 loss $c \in \{0, 1\}$ and prior $P(c) = \frac{1}{2}$. To compute $p(I|c)$, one could either marginalize (MAP) the hidden variables ξ, ν, g, which requires knowledge of the priors $dP(\xi|c)$, $dP(\nu)$ and $dP(g)$, or max-out (ML) the nuisances (i.e. assume uninformative priors).

2.1 Features and templates

MAP and ML require costly computations at decision time, incompatible with real-time operation on a hand-held. Thus, we restrict the family of classifiers to nearest-neighbors and focus on the optimal *representation* $\hat{I}_c$:

$$\hat{c} = \arg\min_{c \in \{0,1\}} d(I, \hat{I}_c) = \|I - \hat{I}_c\|_*$$

The *template* $\hat{I}_c$ can be any statistic (function) of the training data $\{I_k\}_{k=1}^K \sim p(I|c)$. A *feature* $\phi(I)$ is any statistic that does not require label knowledge. The distance $\|\cdot\|_*$ can be defined in terms of a feature, $d(I, \hat{I}_c) \doteq \|\phi(I) - \phi(\hat{I}_c)\|$. This approach does not generally enjoy the properties of the Bayes and ML discriminants [11], so two questions are critical: *What is the "best" template $\hat{I}_c$, and how can it be computed from the training set? Are there conditions when the best template yields optimal classification?* We answer these in order.

2.1.1 What is the "best" template?

The one that induces the smallest expected distance for each class. It depends on the distance function; for the Euclidean case we have $\hat{I}_c = \arg\min_{I_c} E_{p(I|c)}[\|I - I_c\|^2] = \int_{\mathcal{I}} \|I - I_c\|^2 dP(I|c)$, that is solved by the *class-conditional mean* and approximated by the sample mean using the training set

$$\hat{I}_c = \int_{\mathcal{I}} I dP(I|c) \simeq \sum_k h(g_k \xi_k, \nu_k) \tag{2}$$

where the priors $g_k \sim dP(g), \nu_k \sim dP(\nu), \xi_k \sim dQ_c(\xi)$ act as importance distributions. Of course, the averaging operation entails a loss of discriminative power, so the BTD is only "best" among templates. As an alternative, one could retain the distribution aggregated over time, but that would cause the comparison to be more involved. Different instantiations of this approach, corresponding to different groups G, scene models Ξ, and nuisances ν, yield Geometric Blur [9], and DAISY [10]. Rather than designing the priors $dP(g)$, $dP(\nu)$, we will rely on the *active user* to compute the integral in (2) in the training procedure.

In the case of **group nuisances** we can compute the distance on the quotient, $\|I - \hat{I}\|_{\mathcal{I}/G} \doteq \|\phi(I) - \phi(\hat{I})\|$ and avoid blurring-out the group in the template, which yields an optimal (equi-variant) classifier (thm 2). Unfortunately, not all nuisances are groups, a problem we address in sect. 2.2. Until then, we describe how to design features ϕ for group nuisances.

A *feature* $\phi : \mathcal{I} \to \mathbb{R}^F$ (any deterministic function of the data taking values in some vector space) $I \mapsto \phi(I)$ is *G-invariant* if $\phi \circ h(g\xi, \nu) = \phi \circ h(\xi, \nu), \quad \forall\, g \in G$ and ξ, ν in the appropriate spaces. For group nuisances we can define a *complete* (a.k.a. "discriminative" or "sufficient" or "distinctive") feature as one that captures the entire orbit: referring to (1) with $\nu = 0$ (we will address $\nu \neq 0$ in sect. 2.2) we have that $\phi : \mathcal{I} \to \mathbb{R}^{\dim(\Xi)}$ is *complete* iff $[\phi \circ h(g\xi, 0)] \doteq \{g\phi \circ h(\xi, 0),\ \forall\, g \in G\} = [\xi]$. A complete invariant feature is the ideal canonical template, in the sense that it captures everything that is in the data but for the effect of G. Thus we define the canonical representative $\hat{\xi}$ as[1]

$$\hat{\xi} \doteq \phi(I) = \phi \circ h(g\xi, 0) = \phi \circ h(\xi, 0). \tag{3}$$

One of many ways to design an invariant feature is to use the data I to "fix" a particular group element $\hat{g}(I)$, and then "undo" it from the data. If the data does not allow fixing a group element $\hat{g}$, it means it is already invariant to G.

Definition 1 *With reference to (1), a* (G-)co-variant detector *is any functional* $\psi : \mathcal{I} \times G \to \mathbb{R}^{\dim(G)};\ (I, g) \mapsto \psi(I, g)$ *such that (i) The equation* $\psi(I, g) = 0$ *uniquely determines a group element* $\hat{g} = \hat{g}(I)$*, and (ii)* $\psi(I, \hat{g}) = 0$*, then* $\psi(I \circ g, \hat{g} \circ g) = 0\ \forall\, g \in G$*, where* $I \circ g$ *is defined by* $(I, g) = (h(\xi, 0), g) \mapsto h(g\xi, 0) \doteq I \circ g$*.*

The first condition (i) is equivalent to the Jacobian being non-singular:

$$|J_g| \doteq \det\left(\frac{\partial \psi}{\partial g}\right) \neq 0 \tag{4}$$

We say that the image I is *G-canonizable* if there exists a covariant detector ψ such that $\psi(I, \hat{g}) = 0$. Depending on ψ, the statistic may be *local*, i.e. only depend on $I(x), x \in \mathcal{B} \subset \Omega$ on a subset of the image domain $\mathcal{B}$; with an abuse of nomenclature, we say that the *region* $\mathcal{B}$ is canonizable. With a co-variant detector we can easily construct a complete invariant descriptor as follows: For a given co-variant detector ψ that fixes a canonical element $\hat{g}$ via $\psi(I, \hat{g}(I)) = 0$ we call the statistic

$$\phi(I) \doteq \{I \circ \hat{g}^{-1}(I) \ \mid\ \psi(I, \hat{g}(I)) = 0\}. \tag{5}$$

a *canonized descriptor.* The following results are proven in an appendix uploaded as supplementary material.

Theorem 1 (Canonized descriptors are complete invariants) *Let* ψ *be a co-variant detector. Then the corresponding canonized descriptor (5) is a complete invariant statistic.*

Theorem 2 (When is a template optimal?) *If a complete G-invariant descriptor* $\hat{\xi} = \phi(I)$ *can be constructed, a classifier based on the class-conditional distribution* $dP(\hat{\xi}|c)$ *attains the same risk as one based on the likelihood* $p(I|c)$*.*

In the next section we show what groups can be canonized.

2.2 Interaction of invertible and non-invertible nuisances

We now relax the condition $\nu = 0$; the maps $I \circ g \doteq h(g\xi, 0)$ and $I \circ \nu \doteq h(\xi, \nu)$ can be composed, $I \circ g \circ \nu = h(g\xi, \nu)$ but, in general, they do *not* commute. When they do, $I \circ g \circ \nu = I \circ \nu \circ g$, we say that the group nuisance g *commutes* with the (non-group) nuisance ν.

For a nuisance to be canonizable (i.e. eliminated via pre-processing without loss of discriminative power) it has to be invertible *and* commutative. The following theorem, proved in appendix, shows that this is the case only for the isometric group of the plane.

Theorem 3 *The only nuisance that commutes with quantization is the isometric group of the plane (rotations, translations and reflections).*

[1]Note that we drop the subscript c and the superscript from the template since $\phi(\hat{I}_c)$ is invariant to G regardless of the class c, and it is a sufficient statistic, with no approximation when $\nu = 0$.

As a corollary, the *affine group*, and in particular its **scaling** sub-group, can*not* be eliminated in the representation without a loss of discriminative power. This is unlike what [8] prescribes, and [6] uses, since they did not include quantization in their analysis.

Planar **rotations** commute with occlusions and quantization. But, rather than using a co-variant detector as a canonization mechanism [6], we use the *projection of the gravity vector onto the image plane*. While **translation** commutes with quantization, it does *not* commute with occlusion, and therefore it should be marginalized or eliminated at decision time. Following the analysis in [12], with probability one a translation-co-variant detector yields isolated (Morse) critical points $x_i \in \Omega$. Therefore, marginalization/max-out at decision time reduces to a combinatorial hypothesis test (sect. 2.4). In this sense, we say that *translation is locally canonizable*. The next section takes this analysis one step closer to implementation.

2.3 Designing feature detectors

Proper design of a feature detector consists of canonizing the canonizable nuisances in a way that is the least "sensitive" to the non-invertible ones. Sensitivity is traditionally captured by the notion of (BIBO) stability. Unfortunately, this is not meaningful in the context of recognition, and indeed we show in the appendix that *any* co-variant detector as defined in 1 is BIBO stable. Instead, we introduce a different notion of stability that is relevant to recognition [13].

Definition 2 (Structural Stability) *A G-covariant detector $\psi \mid \psi(I, \hat{g}(I)) = 0$ is Structurally Stable if small perturbations $\delta\nu$ preserve the rank of the Jacobian matrix (4):*

$$\exists\ \delta > 0 \mid |J_{\hat{g}}| \neq 0 \Rightarrow |J_{\hat{g}+\delta\hat{g}}| \neq 0 \quad \forall\ \delta\nu \mid \|\delta\nu\| \leq \delta \tag{6}$$

with $\delta\hat{g} \doteq |J_{\hat{g}}|^{-1} \frac{\partial h}{\partial \nu} \delta\nu$.

We define the maximum norm of the nuisance that does not cause a singularity in the detection mechanism the *structural stability margin*, which we use to rank features in sect. 3:

$$\delta^* = \sup \|\delta\nu\| \quad | \quad |J_{\hat{g}+K\delta\nu}| \neq 0 \tag{7}$$

A sound feature detector is one that identifies Morse critical points in G that are as far as possible from singularities. The selection of canonical frames according to this principle is described in the next section.

2.4 Proper Sampling and Correspondence

In traditional signal processing, proper sampling refers to regular sampling at twice the Nyquist frequency. This is irrelevant in recognition, where the task is not to reconstruct an exact copy of some "true" image. A more appropriate condition of proper sampling would be for a feature detector $\psi(I, \hat{g}) = 0$ for the location-scale group $g = \{x, \sigma\}$ to be *topologically equivalent* to the "true" image $\psi(h(\xi, 0), \hat{g}) = 0$.

Unfortunately, we do not have the "true" image $h(\xi, 0)$. However, under the Lambertian assumption, this is equivalent to testing against *the next image* $I_{t+1}(x)$.

Definition 3 (Proper Sampling) *We say that a signal $\{I_t\}_{t=1}^T$ is* properly sampled *at scale σ at time t if the Morse-Smale complex of $(I_t * \mathcal{G}(x; \sigma^2))$ is equal to that of $(I_{t+1} * \mathcal{G}(x; \sigma^2))$.*

Of course, occlusions yield a signal that is *not* properly sampled, which leads to failure of the combinatorial matching test of two local invariant features at decision time, which is what we want. Following standard scale-space theory, in the absence of occlusions, for any signal I, there exists a large enough scale σ such that I is properly sampled at σ. Also, assuming continuity and a sufficiently slow motion relative to the temporal sampling frequency, there exists a large-enough scale σ_{max} such that the video signal is properly sampled at that scale. This is relevant because, typically, temporal sampling is performed at a fixed rate, and we do not want to perform temporal anti-aliasing by artificially motion-blurring the images, as this would destroy spatial structures in the image. Note, however, that once a large enough scale is found, so

correspondence is established at the scale σ_{max}, the motion $\hat{g}_t$ computed at that scale can be compensated, and therefore the (back-warped) images $I_t \circ \hat{g}_t^{-1}$ can now be properly sampled at a scale $\sigma \leq \sigma_{max}$. This procedure can be iterated, until a minimum σ_{min} can be found beyond which no topological consistency is found. Note that σ_{min} may be smaller than the native resolution of the sensor, leading to a *super-resolution* phenomenon.

This analysis is the basis of our integrated approach to selection and tracking, dubbed *tracking on the selection tree* (TST), whereby one first selects structurally stable features via proper sampling. The structural stability margin determines the neighborhood in the next image where *independent selection* is to be performed. If the procedure yields precisely one detection in this neighborhood, topology is preserved, and proper spatio-temporal sampling is achieved. Otherwise, a topological change has occurred, and the track is broken.

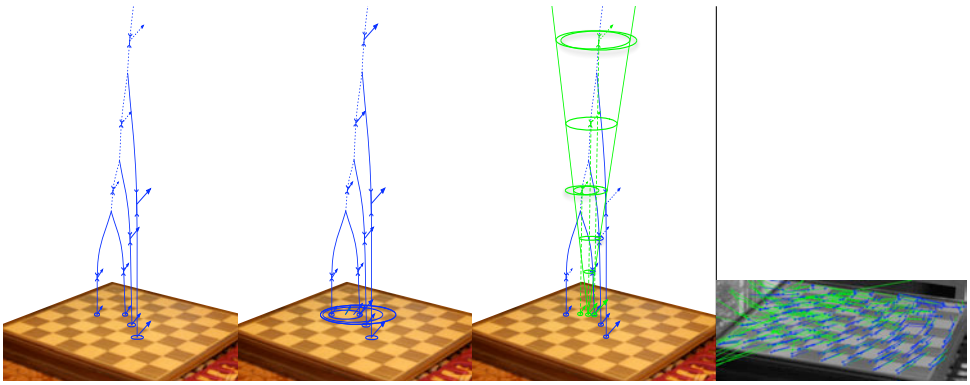

Figure 1: **Tracking on the selection tree.** *The approach we advocate only provides motion estimates at the terminal branches (finest scale); the motion estimated at inner branches is used to back-warp the images so large motion would yield properly-sampled signals at finer scales (left). As an alternative, the motion estimated at inner branches can also be returned, together with their corresponding scale (middle). Traditional multi-scale detection and tracking, on the other hand, first "flattens" all selections down to the finest level (dashed vertical downwards lines), then for all these points consider the entire multi-scale cone above (shown only for one point for clarity). As a result, multiple extrema at* inconsistent *locations in scale-space are involved in providing coarse-scale initialization (right). Motion estimates at a scale* finer *than the native selection scale (thinner green ellipse), rather than improving the estimates, degrade them because of the contributions from spurious extrema (blue ellipses). Motion estimates are shown on the right (blue = TST, green = multi-scale Lucas-Kanade (MLK)).*

This is illustrated in fig. 1. Note that only the terminal branches of the selection scale-space provide an estimate of the frame $\hat{g}$, whereas the hidden branches are used only to initialize the lower branches. Alternatively, one can report each motion estimate at the native selection scale (fig. 1 middle). This is different than multi-scale tracking as traditionally done [4] described in [5] and implemented in OpenCV, illustrated in fig. 1 (right).

2.5 Local invariant frames

The selection procedure yields a topological tree in scale-space with locations $\{x_i\}_{i=1}^N$ and, for each location, multiple scales $\{\sigma_{ij}\}_{i,j=1}^{N,M}$. Once rotation is canonized using gravity as a reference, we have a collection of *similarity (reference) frames* $\hat{g}_{ij} = \{x_i, \sigma_{ij}, R_{ij}\}$ each identifying a region $\mathcal{B}_{\sigma_{ij}}(x - R_{ij}x_i)$, where a complete

contrast invariant can be computed:[2]

$$\phi(I) = \{\frac{\nabla h(\hat{g}_{ij}\xi, \nu)}{\|\nabla h(\hat{g}_{ij}\xi, \nu)\|} = I \circ \hat{g}_{ij}(x) \doteq \phi_{ij}(I) \quad \forall\, x \in \mathcal{B}_{\sigma_{i,j}}(x - R_{i,j}x_i).\}_{i,j=1}^{N,M} \tag{8}$$

The feature $\phi(I)$ is now a multi-component descriptor for the entire image I. Non-invertible nuisances are not canonizable and must be *marginalized* or eliminated at decision time. In particular, *occlusions* are marginalized via a combinatorial matching test of collections of features $\{\phi_{ij}(I)\}$ in different images. Arbitrary *changes of viewpoint* correspond to diffeomorphic domain deformations that do not constrain the frames $\{\hat{g}_{ij}\}$, making the collection $\{\phi_{ij}(I)\}$ a *bag of features*. This in part explains the surprising success of this simplistic model, which we adopt in sect. 3. Feature descriptors computed on a test image must be compared with the best descriptor learned from the training set.

2.6 Learning best-template descriptors

In order to compute the best template (2), one needs to average with respect to the nuisances that have not been canonized. The prior $dP(\nu)$ is generally not known, and neither is the class-conditional density $dQ_c(\xi)$. However, if a sequence of frames $\{\hat{g}_k\}_{k=1}^T$ has been established in multiple *training* images $\{I_k\}_{k=1}^T$, with $I_k = h(g_k\xi_k, \nu_k)$, then it is easy to compute the best (local) template via

$$\hat{I}_c = \int_{\mathcal{I}} I dP(I|c) = \sum_{\substack{\nu_k \sim dP(\nu) \\ \xi_k \sim dQ_c(\xi)}} \phi \circ h(\hat{g}_k\xi_k, \nu_k) = \sum_k I \circ \hat{g}_k = \sum_{k,i,j} \phi_{ij}(I_k) \tag{9}$$

where $\phi_{ij}(I_k)$ are defined in eq. (8) for the k-th image I_k. A sequence of canonical frames $\{\hat{g}_k\}_{i=1}^T$ is the outcome of a *tracking* procedure (sect. 2.4). Note that we are tracking reference frames $\hat{g}_k$, not just their translational component (points) x_i, and therefore tracking has to be performed on the selection tree (fig. 1). *The template above $\hat{I}_c$, therefore, is an averaging of the gradient direction, in a region determined by $\hat{g}_k$, according to the nuisance distribution $dP(\nu)$ and the class-conditional distribution $dQ_c(\xi)$, as represented in the training data.* This *"best-template descriptor"* (BTD) is implemented in sect. 3. It is related to [14, 7, 9, 10] in that it uses gradient orientations, but instead performing spatial averaging by coarse binning, it uses the actual (data-driven) measures and average gradient directions weighted by their standard deviation over time. The major difference is that composing our template *requires local correspondence*, or tracking, of local regions g_k, in the training set.

Note that, once the template descriptor is learned, with the entire scale semi-group spanned in $dP(\nu)$[3] recognition can be performed by computing the descriptors ϕ_{ij} *at a single scale* (that of the native resolution of the pixel). This significantly improves the computational speed of the method, which in turn enables real-time implementation on a hand-held device (sect. 3).

2.7 Learning priors (and categories)

Instead of having to learn the priors for each object separately during training, we can exploit the training of multiple objects to learn priors that can be shared among objects or categories. Assuming canonizable nuisances have been eliminated (although this is not strictly necessary, hence we will maintain the notation g, ν for all nuisances), the learning procedure consists in solving, to the extent possible, for the model parameters

$$\hat{\xi}, \hat{g}_k, \hat{\nu}_k = \arg\min_{\xi, g_k, \nu_k} \|I_k - h(g_k\xi, \nu_k)\|_* \tag{10}$$

[2] Alternative contrast-invariant mechanisms include local contrast normalization or spectral ratios computed from color images.

[3] Either because of a sufficiently rich training set, or by extending the data to a Gaussian pyramid in post-processing.

The problem (10) can be shown to be equivalent (under the Lambertian assumption) to image-to-image matching as described in sect. 2.4. Once TST has been performed (yielding $\hat{g}_i$), and the residual computed (yielding $\hat{\nu}_i$), sample-based approximations for the nuisance distributions can be obtained, for instance

$$dP(\nu) = \sum_i \kappa_\nu(\nu - \hat{\nu}_i)d\mu(\nu);\ dP(g) = \sum_i \kappa_g(g - \hat{g}_i)d\mu(g); \tag{11}$$

where κ are suitable kernels (Parzen windows). If the problem cannot be solved uniquely, for instance because there are entire subsets of the solution space where the cost is constant, this does not matter as any solution along this manifold will be valid, accompanied by a suitable prior that is uninformative along it.

When the class is represented *not* by a single template ξ, but by a distribution of templates, as in *category recognition*, the problem above can be generalized in a straightforward manner, yielding a solution $\hat{\xi}_i$ at each capture session, from which a class-conditional density can be constructed.

$$dQ_c(\xi) = \sum_{i=1}^{M} \kappa_\xi(\xi - \hat{\xi}_i)d\mu(\xi). \tag{12}$$

An alternative to approximating the density $Q_c(\xi)$ consists of keeping the entire set of samples $\{\hat{\xi}\}$, or grouping the set of samples into a few statistics, such as the modes of the distribution dQ_c, for instance computed using Vector Quantization, which is the choice we adopt in our implementation in sect. 3.

3 Implementation on an iPhone

We have implemented the recognition system described above on an iPhone 3GS with a 600MHz ARM chip CPU. Each image is captured sequentially with a refresh rate of 15 frames-per-second (FPS). The screen is split into two, half for capture half for visualization, resulting in a spatial sampling of 320×240 pixels.

3.1 Feature Detection and Tracking

To determine the correspondence of (canonical reference) frames $\hat{g}_{ij}(t)$ as described in sect. 2.4, for each scale $\sigma_j,\ j = 0, \dots, 4$, limited by computational resources, we perform independent detection of $x_i(t)$ as in sect. 2.5 using FAST corner detection [14] with size and threshold parameters 9 and 20 respectively, with non-maximal suppression to guarantee proper (spatial) sampling as described in sect. 2.4. The rotational reference R_{ij} can be fixed by gravity as described in sect. 2.5 or assumed vertical. Each feature $\hat{g}_{ij} \doteq \{x_i, \sigma_j, R_{ij}\}$ is scored in decreasing order of *structural stability* from def. 2, by measuring the scale-normalized distance to the nearest detected feature. Correspondence is established for the translational component $x_i(t+1)$ via a simple (differential) translational tracking algorithm [4] that, starting from the locations selected at the coarsest scale $j = 4$, provides $v_{i4}(t)$ such that $\hat{x}_i(t+1) \doteq x_i(t) + v_{i4}(t)$ for all $x_i(t)$ selected at scale $j = 4$. The new image is then back-warped by $-2v_{i4}(t)$ in each region $\mathcal{B}_{\sigma_3}(x - x_i(t+1))$, described in eq. (8). There, we re-select points $x_i(t+1)$ within the back-warped region, and repeat the procedure as described in sect. 2.4.

If a topological change occurs at level j, the motion v_{ij} is *not* propagated to level $j-1$, and is instead reported as a motion estimate for x_i with native scale σ_j. From level $j-1$ onward, the (multiple, or none) features x_i that fall within $\mathcal{B}_{\sigma_{j+1}}(x - x_i(t+1))$ are used to propagate velocity estimates down until $j = 0$, as illustrated in fig. 1 (left). In order to keep the number of tracked features between 40 and 50, rather than only reporting motion at the finest scale v_{i0}, we report motion at all scales, v_{ij}, each with its own scale σ_j, as illustrated in fig. 1 (middle).

This approach differs from traditional multi-scale feature detection and tracking as described in sect. 2.3. It enables tracking over relatively large baselines as shown empirically in fig. 1, and improves accuracy and (structural) stability, defined in def. 2 and quantified by the number of inlier matches. A quantitative experiment on real sequences is reported in sect. 3.4 and demonstrated in a video clip uploaded as supplementary material.

Although ideally we would like to have a full geometrically-validated outlier rejection stage [15], the iPhone does not enable this to run in real-time. Therefore, we have settled for a coarser hypothesis test where the (two-dimensional) configuration of selected features is hypothesized constant in the absence of occlusions, in a coarsely binned histogram. Given a tracked feature, if its neighboring features in the previous frame are tracked as its neighbors in the current frame again with more than a 0.5 threshold ratio, we classify it as inlier, and otherwise reject it as a (partial) occlusion. Feature tracking results are then used to limit the search space for feature detection in the new image. This is in line with a diffeomorphic domain deformation model (sect. 2.5).

3.2 Feature Descriptors

Once local frames $\hat{g}_{ij}$ are available, we compute descriptors around each one following the guidelines of sect. 2.6. For each selected and tracked region, we compute gradient orientation using portions of source code from the VLFeat library. Instead of building the scale-space for the original SIFT algorithm, we use the image pyramid available from feature detection and tracking. In our implementation we have tested both the standard SIFT descriptor, and the BTD with local contrast normalization (sect. 2.6). Both are updated periodically as long as their corresponding frame is being tracked. Due to computational constraints, we limit the number of SIFT descriptors computed at each frame to 5, selected among the features being tracked, whereas the BTD can be computed for every features.

As discussed in sect. 2.5, the scale semi-group can be computed (off-line) after training data is available. We limit the on-line description generation to the native scale of the images being captured, and defer the scaling process off-line.

In each case, the descriptors are quantized using a vocabulary tree using hierarchical K-means [16]. We used the training images from the 2009 PASCAL [1], and extracted 1M descriptors. The vocabulary tree is built with 4 levels and 8 clusters each, forming 4096 clusters with centers in the leaf nodes. Thus each descriptor can be represented as a short integer.

3.3 Recognition in a single video frame

Once a template is learned from multiple video frames, recognition is possible from a single image. We use standard methods consisting of a bag-of-features model of features ϕ_{ij} described in eq. (8), compatible with an arbitrary viewpoint change for objects of general shape as described in sect. 2.5. The quantized descriptors are used for learning object models, and also for recognizing the objects in a video frame. For scoring a set of features with respect to a certain object, we use a term frequency-inverse document frequency (TF-IDF) scheme, modified by substituting $\#(\phi_{ij}, d)$, the number of features ϕ_{ij} corresponding to an object, with either 1 or 0, depending on whether the corresponding feature is present or not. This way, a user can take multiple views of an object effectively while learning the model of the object without producing skewed sampling of features from different views. To recognize an object, we compute the TF-IDF score of the set of features and compare against all the learned object models, and choose the object with the highest score.

While multiple video frames are indispensable in training, as described in sect. 2.7, they are not strictly necessary for recognition. In our current implementation we perform independent classification for each image. However, one could treat each image as a weak classifier in a cascade-of-classifier framework [17]. Although this is conceptually straightforward, it adds complexity to the process and has therefore not been implemented in the current release.

3.4 Performance

The performance of our system, tested off-line, is qualitatively comparable with algorithms performing at baseline levels on standard datasets such as the Caltech 101. However, direct comparison is not straightforward because we do not use multiple (supervised) hand-labeled training samples for each category, but instead use multiple images of the *same* object, relying on the user to sample multiple aspects (viewpoints). Although one may argue that our model is for *individual object* recognition, not object categories, the

performance in recognizing object classes represented in the Caltech 101 dataset is similar to the baseline algorithms.

To obtain more meaningful performance evaluation, we compare the individual elements of our system with the equivalent modules commonly used in the literature.

We compare TST tracking with standard multi-scale Harris corner selection with multi-scale Lucas-Kanade tracker, as implemented in the OpenCV, which we refer to as MLK. Representative experiments are illustrated in fig. 1, and quantitative experiments are reported in fig. 2 and table 1. There, it can be seen that our approach is faster (for an equal number of tracked features), more accurate (a smaller median motion error), and considerably more robust (a smaller spread between the mean and the median). With this, the overall recognition that involves capturing images, detecting and tracking features, and calculating descriptors is performed at a rate of about 7 frames-per-second.

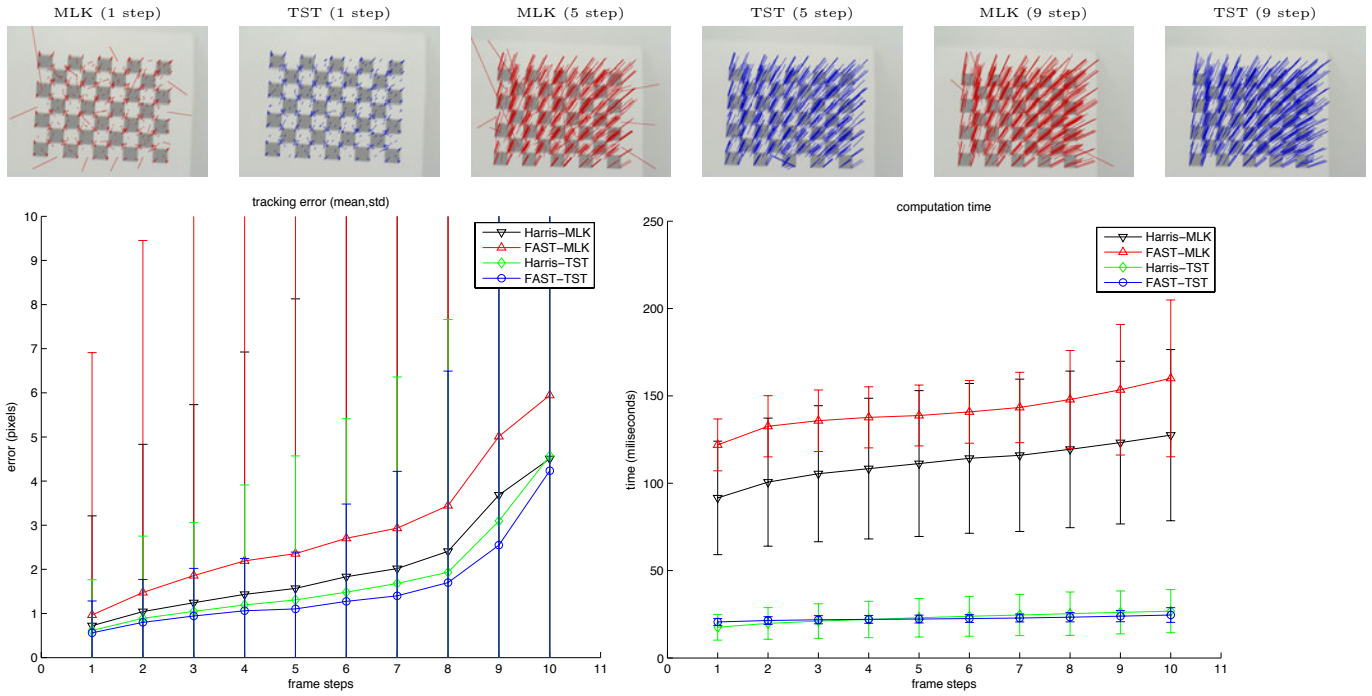

Figure 2: **Comparison of multi-scale translational tracking and tracking on the selection tree.** *First row: qualitative comparison of TST and MLK with increasing parallax. For a quantitative comparison, Harris corners and FAST corners are tested for both TST and MLK. Bottom-left: tracking error for these combinations of methods. Right: computation time. FAST-TST performs best in both accuracy and speed.*

In fig. 3 we illustrate the comparison of the best-template descriptor with an equivalent SIFT descriptor. To compare the two, we use a small set of objects (5-10) from [18]. An exhaustive experimental comparison is not straightforward since the BTD depends on the training set, unlike other descriptors cited. Therefore, we can construct cases with rich training sequences where the BTD outperforms all other descriptors, even if they are learned on the same set (because we have the advantage of fine correspondence), and vice-versa we can construct poor training sequences where a straight bag of SIFT features computed independently in each video frame outperforms our approach. Representative quantitative experiments are reported in fig. 3 where it can be seen that SIFT and BTD perform similarly in terms of accuracy, but SIFT is significantly more costly to compute. An exhaustive experimental comparison represents a separate contribution and is currently under development. Qualitative results are illustrated in videos uploaded as supplementary material.

	tracking error (pixels)			computation time (ms)	inlier ratio (%)
	median	mean	std.		
Harris-MLK short	0.35	0.72	2.49	91.62	83.18
FAST-MLK short	0.36	0.97	5.94	121.95	80.69
Harris-TST short	0.35	0.61	1.15	**17.58**	84.31
FAST-TST short	**0.34**	**0.56**	**0.72**	20.60	**85.74**
Harris-MLK long	0.83	2.19	9.30	114.00	59.26
FAST-MLK long	0.88	3.10	14.66	143.39	55.85
Harris-TST long	0.84	1.91	5.37	23.63	58.17
FAST-TST long	**0.81**	**1.67**	**4.00**	**22.78**	**59.28**

Table 1: *TST and MLK are compared on short baseline (two adjacent frames) and long baseline (skipping every two or more frames).*

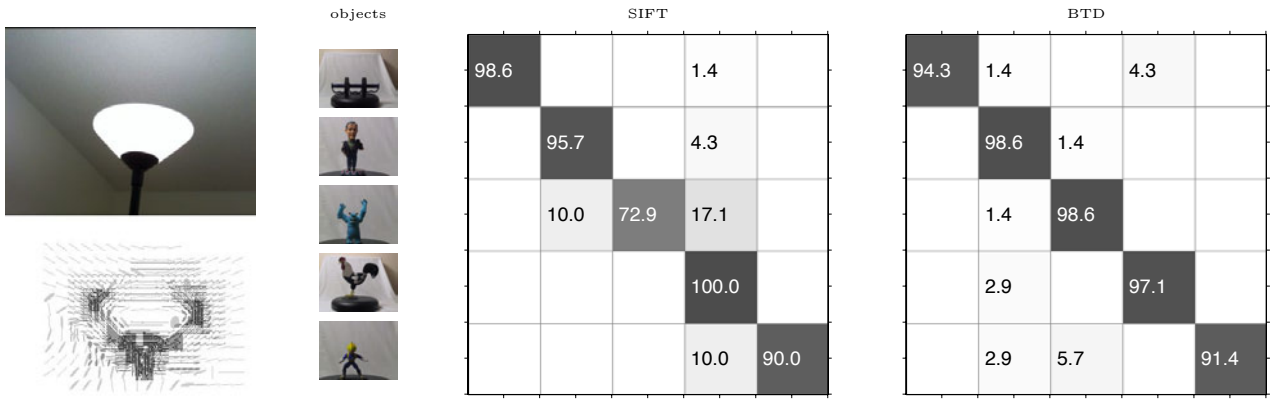

Figure 3: **Comparing SIFT and best-template descriptors.** *A representative sample of the BTD is shown in the left. We generate confusion matrices of using SIFT and BTD, by training objects at one scale and testing for different viewing angles and scales [18]. Performance is similar (some trials go in favor of SIFT, others to BTD, depending on the training sequences), but BTD is significantly faster to compute, as shown in table 2.*

	Tracking	Descriptors	Frame Rate
TST-BTD	40 ms	15 ms	7 fps
MLK-SIFT	100 ms	180 ms	3 fps

Table 2: *Runtime computation time on an iPhone.*

4 Discussion

We have described an implementation of a recognition system on a mobile device and an analysis that motivates the design choices in light of attempting to make the run-time cost of the algorithm as small as possible. By restricting the classifier to a simple comparison to a *template*, we are forced to consider which one is the best template. This leads to the *best-template descriptor* (BTD).

The need to integrate correspondence, or tracking, into recognition forces us to implement an efficient feature selection and tracking mechanism. Guided by the notion of (Morse) isolation and proper spatio-temporal sampling, we have designed a modified (similarity)-frame detection and tracking algorithm, TST. It is cheaper and better than stock algorithms available, for instance, through the OpenCV and VLFeat software libraries.

Once a representation is in place, we use standard clustering and scoring algorithms to perform recognition. At present, each captured objects represents a category in itself, and the score is visualized on the graphic display.

Our BTD assumes that the object is (at least locally) rigid, and domain deformation is due to changes of viewpoint. Thus it is not suited for complex articulated objects such as humans. In that case, the blurred template will lose discriminative power. Instead, one would need to independently track and describe rigid parts, and group them as a coherent entity in post-processing. This is a research program in itself beyond the scope of this paper.

The algorithm we propose can be easily implemented by the skilled reviewer on a simulation platform. The system implementation on an iPhone will be available for free download, and a video describing its operation is uploaded as supplementary material.

References

[1] Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The PASCAL Visual Object Classes Challenge 2009 (VOC2009) Results. http://www.pascal-network.org/challenges/VOC/voc2009/workshop/ (2009)

[2] Shotton, J., Johnson, M., Cipolla, R., Center, T., Kawasaki, J.: Semantic texton forests for image categorization and segmentation. In: IEEE CVPR. (2008) 1–8

[3] Salton, G., McGill, M.: Introduction to modern information retrieval. McGraw-Hill New York (1983)

[4] Lucas, B., Kanade, T.: An iterative image registration technique with an application to stereo vision. In: Image Understanding Workshop. (1981) 121–130

[5] Baker, S., Matthews, I.: Equivalence and efficiency of image alignment algorithms. In: IEEE CVPR. Volume 1. (2001) 1090–1097

[6] Lowe, D.G.: Distinctive image features from scale-invariant keypoints. IJCV **2** (2004) 91–110

[7] Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: IEEE CVPR. (2005) 886–893

[8] Lindeberg, T.: Principles for automatic scale selection. Technical report, KTH, Computational Vision and Active Perception laboratory (1998)

[9] Berg, A., Malik, J.: Geometric blur for template matching. In: IEEE CVPR. (2001) 607

[10] Tola, E., Lepetit, V., Fua, P.: A fast local descriptor for dense matching. In: IEEE CVPR. (2008) 1–8

[11] Robert, C.P.: The Bayesian Choice. Springer Verlag, New York (2001)

[12] Mumford, D., Gidas, B.: Stochastic models for generic images. Quarterly of Applied Mathematics **54** (2001) 85–111

[13] Matas, J., Chum, O., Urban, M., Pajdla, T.: Robust wide baseline stereo from maximally stable extremal regions. In: BMVC. (2002)

[14] Rosten, E., Drummond, T.: Machine learning for high-speed corner detection. In: ECCV. Volume 1. (2006) 430–443

[15] Fischler, M.A., Bolles, R.C.: Random sample consensus: a paradigm for model fitting with application to image analysis and automated cartography. Comm. of ACM **24** (1981) 381–395

[16] Nister, D., Stewenius, H.: Scalable recognition with a vocabulary tree. In: IEEE CVPR. Volume 2. (2006) 2161–2168

[17] Viola, P., Jones, M.: Robust real-time object detection. In: Second International Workshop on Statistical and Computational Theories of Vision. (2001)

[18] Moreels, P., Perona, P.: Evaluation of features detectors and descriptors based on 3d objects. IJCV **73** (2007) 263–284

SUPPLEMENTARY MATERIAL

This appendix describes the video uploaded, and collates some support material referenced in the text. The latter can be skipped without compromising the integrity of our contribution.

A Uploaded video

The first part of the video shows a representative sample of a tracking sequence, whereby the iPhone is moved at faster and faster speed in front of a complex scene, that includes a checkerboard. OpenCV's camera calibration method for checkerboard patterns is used to generate "ground truth" values for the quantitative comparisons reported in the paper. It can be seen that the TST is fast and reasonably accurate even for fast motions, since it is designed to be maximally structurally stable. This algorithm improves stock approaches such as MLK both in terms of accuracy, robustness, and computational speed.

The second part shows the user interface of the recognition system. The user selects an object by clicking on the screen, and moves the phone to learn a model (BTD). The process is repeated for a number of objects usually ranging between 5 and 10. If the user wants to group multiple objects under the same label, he or she can drag the icon of the newly learned object onto that of an existing one. In recognition mode, the system displays the icon of the object being recognized, based on highest TF-IDF score, in real-time for every captured frame.

B BIBO Stability

As mentioned in the paper, the traditional notion of stability, that measures the sensitivity of a descriptor with respect to small perturbations of a nuisance, is irrelevant to recognition. Indeed, we now show that any properly designed co-variant detection is automatically stable in this sense.

Definition 4 (BIBO stability) *A G-covariant detector ψ (Def. 1) is* bounded-input bounded-ouput *(BIBO) stable if small perturbation in the nuisance cause small perturbations in the canonical element. More precisely, $\forall\ \epsilon > 0\ \exists\ \delta = \delta(\epsilon)$ such for any perturbation $\delta\nu$ with $\|\delta\nu\| < \delta$ we have $\|\delta\hat{g}\| < \epsilon$.*

Note that $\hat{g}$ is defined implicitly by the functional equation $\psi(I, \hat{g}(I)) = 0$, and a nuisance perturbation $\delta\nu$ causes an image perturbation $\delta I = \frac{\partial h}{\partial \nu}\delta\nu$. Therefore, we have from the Inverse Function theorem[4]

$$\delta\hat{g} = -|J_{\hat{g}}|^{-1}\frac{\partial h}{\partial \nu}\delta\nu \doteq K\delta\nu \tag{13}$$

where J_g is the Jacobian (4) and K is called the *BIBO gain.* As a consequence of the definition, $K < \infty$ is finite. The BIBO gain can be interpreted as the sensitivity of a detector with respect to a nuisance. Most existing feature detector approaches are BIBO stable with respect to simple nuisances. Indeed, we have the following

Theorem 4 (Covariant detectors are BIBO stable) *Any covariant detector is BIBO-stable with respect to noise and quantization.*

BIBO stability is reassuring, and it would seem that a near-zero gain is desirable, because it is "maximally (BIBO)-stable." However, simple inspection of (13) shows that $K = 0$ is not possible without knowledge of the "true signal." In particular, this is the case for quantization, when the operator ψ must include spatial averaging with respect to a shift-invariance kernel (low-pass, or anti-aliasing, filter). However, *a non-zero BIBO gain is irrelevant for recognition,* because it corresponds to an additive perturbation of the domain deformation (domain diffeomorphisms are a vector space), which is a nuisance to begin with. On the other hand, structural instabilities are the plague of feature detectors.

[4]One has to exercise come care in defining the proper (Frechèt) derivatives depending on the function space where ψ is defined.

C Proofs

Below are the proofs of the claims made in the paper.

Proof of thm 1: *To show that the descriptor is invariant we must show that* $\phi(I \circ g) = \phi(I)$. *But* $\phi(I \circ g) = (I \circ g) \circ \hat{g}^{-1}(I \circ g) = I \circ g \circ (\hat{g}g)^{-1} = I \circ g \circ g^{-1}\hat{g}^{-1}(I) = I \circ \hat{g}^{-1}(I)$. *To show that it is complete it suffices to show that it spans the orbit space* $\mathcal{I}/G$, *which is evident from the definition* $\phi(I) = I \circ g^{-1}$.

Proof of thm 2: The proof follows from the definitions and theorem 7.4 on page 269 of [11].

Proof of thm 3: *We want to characterize the group* g *such that* $I \circ g \circ \nu = I \circ \nu \circ g$ *where* ν *is quantization. For a quantization scale* σ, *we have the measured intensity (irradiance) at a pixel* x_i

$$I \circ \nu(x_i) \doteq \int_{\mathcal{B}_\sigma(x_i)} I(x)dx = \int \chi_{\mathcal{B}_\sigma(x_i)}(x) I(x)dx \doteq \int \mathcal{G}(x - x_i; \sigma) I(x)dx \tag{14}$$

where $\mathcal{B}_\sigma(x)$ *is a ball of radius* σ *centered at* x, χ *is a characteristic function that is written more generally as a kernel* $\mathcal{G}(x; \sigma)$, *allowing the possibility of more general quantization or sampling schemes, including soft binning based on a partition of unity of* Ω *rather than simple functions* χ. *Now, we have*

$$(I \circ \nu) \circ g(x_i) = \left(\int \mathcal{G}(x - x_i; \sigma) I(x)dx \right) \circ g = \int \mathcal{G}(x - gx_i; \sigma) I(x)dx \tag{15}$$

whereas, with a change of variable $x' \doteq gx$, *we have*

$$(I \circ g) \circ \nu(x_i) = \int \mathcal{G}(x - x_i; \sigma) I(gx)dx = \int \mathcal{G}(g^{-1}(x' - gx_i); \sigma) I(x') |J_g| dx' \tag{16}$$

where $|J_g|$ *is the determinant of the Jacobian (4) computed at* g, *so that the change of measure is* $dx' = |J_g|dx$. *From this it can be seen that the group nuisance commutes with quantization if and only if*

$$\begin{cases} \mathcal{G} = \mathcal{G} \circ g \\ |J_g| = 1. \end{cases} \tag{17}$$

That is, the quantization kernel has to be G*-invariant,* $\mathcal{G}(x; \sigma) = \mathcal{G}(gx; \sigma)$, *and the group* G *has to be an* isometry. *The only isometry of the plane is the set of planar rotations and translations (the Special Euclidean group* $SE(2)$*) and reflections. The set of isometries of the plane is often indicated by* $E(2)$.

Proof of thm 4: *Noise and quantization are additive, so we have* $\frac{\partial h}{\partial \nu} \delta\nu = \delta\nu$, *and the gain is just the inverse of the Jacobian determinant,* $K = |J_{\hat{g}}|^{-1}$. *Per the definition of co-varianat detector, the Jacobian determinant is non-zero, so the gain is finite.*

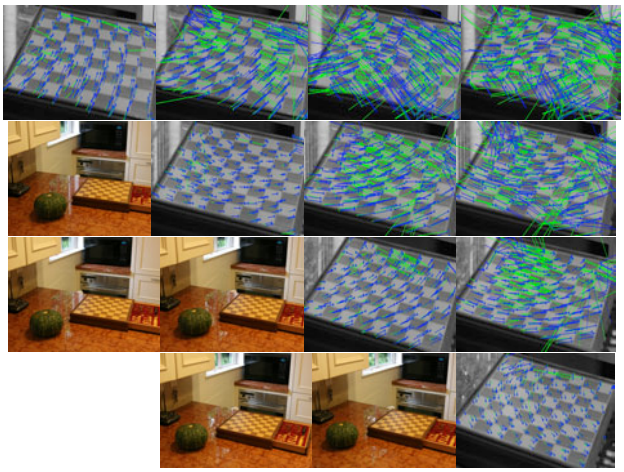

Figure 4: **Comparison of multi-scale translational tracking and tracking on the selection tree** *A representative example of the performance of our approach (blue arrows) compared to standard multi-scale feature detection and tracking (green arrows). Sample images are from many sequences are shown in color (left), and the corresponding estimated displacements are arranged in a matrix, where the position indicates the source images: Images along the diagonal indicate displacement between images 1 and 2, 2 and 3, 3 and 4, 4 and 5 respectively. Images above the diagonal show the estimated motion between images 1 and 3, 2 and 4, 3 and 5; further up for images 1 and 4, 2 and 5, and finally 1 and 5. Significant mismatching occurs as a result of tracking at all scales features that only exist at some scales (green). Restricting multi-scale tracking to matched-scales determined at selection reduces the ambiguities, and is also significantly faster:* 10.49 *FPS for tracking the same number of features on the selection tree (blue) vs.* 4.67 *FPS for tracking in the standard pyramid (green). Quantitative experiments are reported in table 3. Note that the discrepancy between the two methods increases with the baseline: For small baseline (images along the diagonal), they are similar (the blue arrows are drawn on top of the green ones), but for larger disparities (images above the diagonal) the discrepancy grows.*

	median % error	mean % error	std.	time [ms]/ # pts
MLK short	7.55	40.41	142.24	133.24ms
TST short	**5.89**	19.87	52.50	**16.16**ms
MLK < 10%	3.73	4.24	2.72	822# (59.7%)
TST < 10%	**3.65**	4.08	2.55	**926# (67.3%)**
MLK long	127.17	358.17	874.65	188.53ms
TST long	**14.78**	156.72	631.62	**26.98**ms
MLK < 10%	**3.17**	3.85	2.58	645# (32.5%)
TST < 10%	3.90	4.24	2.52	**822# (41.4%)**

Table 3: *TST improves over multi-scale tracking in both accuracy, robustness and speed. The median, mean and standard deviation of the error, normalized as a percentage of the ground-truth velocity vector (to take into account images of different resolution) is reported above. The top block, "short," refers to an short-baseline, where the motion of a checkerboard moving in free-space (fig. 4) is estimated between adjacent video frames (frames 1-2, 2-3, 3-4, etc.). Both algorithms perform reasonably well under these conditions, with an improvement of about 20% in the median error and over 700% in computation speed for the TST approach. The large mean and standard deviation of MLK indicates a high percentage of outliers and mismatches. This is reflected in the second block, where only features that returned motion estimates with less than 10% error are accounted for. In this case, the algorithms perform similarly, but TST has a larger number of valid features (a 12% improvement). The lower block shows the same results for a "long" baseline experiment, where the motion is estimated between video frames separated by 2, 3, and 4 steps (frames 1-3, 2-4, . . . , 1-3, 2-5, . . . , 1-4, 2-5, . . . etc.). Again, when all features are accounted for, TST shows marked improvement over MLK. When only the results with less than 10% error are counted, MLK slightly outperforms TST (by 18%), but uses significantly fewer features (22% less). Of course, in the absence of ground truth one does not know which features perform in the top 10%, and therefore, in the absence of sophisticated and time-consuming robust matching tests, the overall performance of all the features is most important. In this case, TST outperforms MLK by a large margin (over 700% in accuracy and 600% in speed). Note that the results are averaged over a number of runs, so the number of inliers is aggregated over a number of experimental runs, shown with the ratio to the total number of tracked features.*