Robust Tracking in Low Light and Sudden Illumination Changes

Hatem Alismail, Brett Browning, Simon Lucey
Robotics Institute
Carnegie Mellon University
{halismai,brettb,slucey@cs.cmu.edu}

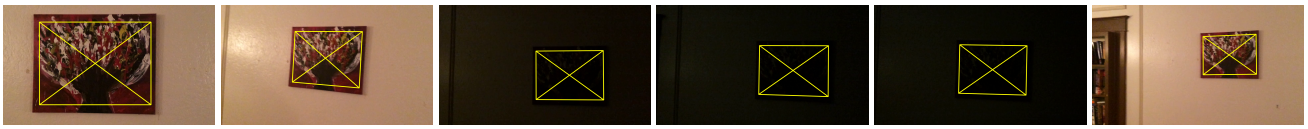
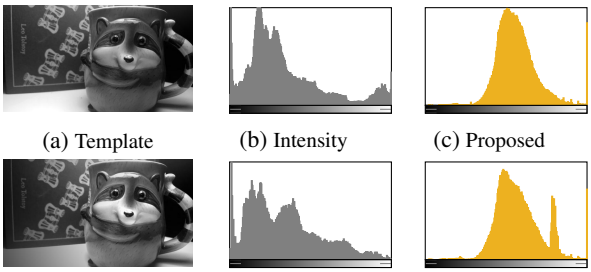


Figure 1: Illustration of tracking robustness in low light and under sudden and drastic illumination changes.

Abstract

We present an algorithm for robust and real-time visual tracking under challenging illumination conditions characterized by poor lighting as well as sudden and drastic changes in illumination. Robustness is achieved by adapting illumination-invariant binary descriptors to dense image alignment using the Lucas and Kanade algorithm. The proposed adaptation preserves the Hamming distance under least-squares minimization, thus preserving the photometric invariance properties of binary descriptors. Due to the compactness of the descriptor, the algorithm runs in excess of 400 fps on laptops and 100 fps on mobile devices.



(a) Template (b) Intensity (c) Proposed

(d) Input (e) Intensity (f) Proposed

Figure 2: Appearance variations between images may appear subtle. But, closer inspection of the distribution of intensities reveals otherwise. The proposed descriptor maintains its distribution shape as a function of nonlinear changes in illumination.

1. Introduction

Visual tracking is an important problem with myriad applications in Vision, and Robotics. With the availability of high frame-rate data, or equivalently small inter-frame displacements, dense methods for image registration exemplified by the Lucas and Kanade algorithm (LK) [27] and Horn and Schuck [21] are the methods of choice for efficient subpixel tracking. Dense, or direct, methods enjoy enhanced precision as most of the image pixels are used to estimate a small number of degrees-of-freedom [23], and are also efficient and amenable to parallelization [24]. Nonetheless, dense methods rely on the *brightness constancy* assumption requiring constant reflection as a function of changing illumination, which is seldom satisfied in real scenarios as illustrated in Figs. 1 and 2.

Developing illumination-invariant tracking algorithms remains an important research problem as evident by the range of algorithms in the literature, where two main paradigms for robust tracking can be found. The first is formulating the tracking objective using intrinsically robust cost metrics [34] such as the Mutual Information [13, 32], or the Normalized Correlation [22, 14], which have been shown to perform robustly especially for multi-modal data [12]. The second approach relies on estimating the illumination parameters alongside the motion [4, 2, 44].

On the one hand, robust cost metrics are more sensitive to the initialization point and their optimization is more challenging than least-squares as accurate estimates of second-order derivatives (the Hessian) are typically required [29]. Conversely, estimating the illumination parameters relies on modeling assumptions, which are difficult to generalize and craft correctly.

In this work, we propose densely evaluated local feature descriptors as a nonparametric means for robust illumination invariant tracking. In particular, we demonstrate the integration of binary descriptors [43] in a multi-channel LK

framework for robust and real-time performance.

Feature descriptors have been instrumental in the evolution of computationally efficient sparse image alignment algorithms. In particular, binary descriptors promise enhanced robustness due to their invariance to various nonlinear intensity deformations [19, 15, 17, 40]. Binary descriptors are also efficient to compute, especially in hardware as floating-point operations are typically not required.

The use of the descriptor constancy in direct tracking is relatively new and is beginning to be explored [37, 1, 6, 10]. Among the first applications of descriptors in tracking is the "distribution fields" approach, which targeted the preservation of local image details at coarse octaves of the scale-space [36]. In optical flow, descriptor matching constraints have been integrated to address large motions [8]. Recently, the use of dense HOG [11] and SIFT [26] has been demonstrated for estimating correspondences across object categories using the LK algorithm [6]. Closest to our work are the "descriptor fields" tracking approach (DF) [10] and the feature-based LK algorithm (FLK) [1]. Descriptors in the DF approach are computed by separating a smoothed version of the image gradient into positive and negative signals [10]. In FLK [1], a number of feature descriptors have been used in tracking and Active Appearance Models [9].

To date, however, the use of binary features in tracking has been limited to pixel accuracy [39, 28], which is often insufficient for accurate tracking. In this work, we demonstrate the use of binary descriptors for sub-pixel alignment tasks using the LK algorithm. Critical to maintaining the invariance of binary descriptors is matching them under appropriate binary norms such as the Hamming distance [5]. Nonetheless, binary metrics are often non-differentiable and are usually approximated [40]. The approximation of the matching metric, however, comes at the price of reduced photometric invariance. A notable example of binary features within gradient-based optimization has been demonstrated on face alignment tasks [33]. But, the approach remains limited to sparse facial landmarks.

Unique to our adaption of binary features for LK is the equivalence of the sum of squared residuals to the Hamming distance as we illustrate in Section 3. In the next section, we review the multi-channel LK formulation.

2. Multi-Channel Lucas-Kanade

Let $\mathbf{I}_0 : \mathbb{R}^2 \to \mathbb{R}$ be the template/reference image. After camera motion with parameter vector $\boldsymbol{\theta} \in \mathbb{R}^p$, we obtain an input/moving image $\mathbf{I}_1$. We desire to estimate the parameters of motion such that we minimize:

$$\mathcal{E}(\mathbf{x};\boldsymbol{\theta}) = \sum_{\mathbf{x}\in\Omega_0} \|\mathbf{I}_0(\mathbf{x}) - \mathbf{I}_1(\mathbf{x}'(\boldsymbol{\theta}))\|_2^2, \qquad (1)$$

where Ω_0 is a subset of pixels in the template, $\boldsymbol{\theta}$ is an initial estimate of the motion parameters and $\mathbf{x}'(\boldsymbol{\theta})$ describes the transformed pixel coordinates given the motion parameters, commonly known as the *warping* function. By performing a first-order Taylor expansion of Eq. (1) in the vicinity of $\boldsymbol{\theta}$, taking the derivative with respect to the parameters, and equating it to zero, we arrive at the normal equations: $\mathbf{J}(\mathbf{x};\boldsymbol{\theta})^\top \mathbf{J}(\mathbf{x};\boldsymbol{\theta}) \Delta\boldsymbol{\theta} = \mathbf{J}(\mathbf{x};\boldsymbol{\theta})^\top \mathbf{e}(\mathbf{x};\boldsymbol{\theta})$, where $\mathbf{J}(\mathbf{x};\boldsymbol{\theta})$ is the matrix of partial derivatives of the warped image intensities with respect to the motion parameters evaluated at the current estimate of parameters $\boldsymbol{\theta}$, and $\mathbf{e}(\mathbf{x};\boldsymbol{\theta}) = \mathbf{I}_0(\mathbf{x}) - \mathbf{I}_1(\mathbf{x}'(\boldsymbol{\theta}))$ is the vector of residuals. Using the chain rule, we obtain $\mathbf{J}(\mathbf{x};\boldsymbol{\theta}) = \frac{\partial \mathbf{I}_1(\mathbf{x})}{\partial \boldsymbol{\theta}} = \frac{\partial \mathbf{I}}{\partial \mathbf{x}'}\frac{\partial \mathbf{x}'}{\partial \boldsymbol{\theta}}$, where $\partial \mathbf{I}_1 / \partial \mathbf{x}'$ is estimated stochastically through x- and y-finite differences, while $\partial \mathbf{x}' / \partial \boldsymbol{\theta}$ is usually obtained deterministically using the closed-form of the warping function. The original formulation of LK is applicable to a variety of problems. For special warps that satisfy a group requirement, however, a more efficient variant is Baker & Matthews' Inverse Compositional algorithm (IC) [3] which we will use in the experimental portion of this paper.

The extension to multi-channels proceeds as follows. Let $\boldsymbol{\phi}_0 : \mathbb{R}^2 \to \mathbb{R}^d$ be the d-channel representation of the template image. Employing a similar notation to the classical LK algorithm, after camera motion with parameter vector $\boldsymbol{\theta} \in \mathbb{R}^p$, we obtain an input d-channel representation $\boldsymbol{\phi}_1$. To align descriptors using LK we seek to minimize:

$$\mathcal{E}_{\boldsymbol{\phi}}(\mathbf{x};\boldsymbol{\theta}) = \sum_{\mathbf{x}\in\Omega_0} \sum_{j=1}^{d} \|\boldsymbol{\phi}_0^j(\mathbf{x}) - \boldsymbol{\phi}_1^j(\mathbf{x}'(\boldsymbol{\theta}))\|^2, \qquad (2)$$

where the j^{th} channel is denoted with $\boldsymbol{\phi}^j$ such that $\boldsymbol{\phi}(\mathbf{x}) = [\boldsymbol{\phi}^1(\mathbf{x}) \dots \boldsymbol{\phi}^d(\mathbf{x})]^\top$. To linearize Eq. (2) we must obtain an estimate of the Jacobian $\mathbf{J}_{\boldsymbol{\phi}}(\mathbf{x};\boldsymbol{\theta}) = \partial \boldsymbol{\phi} / \partial \boldsymbol{\theta} \in \mathbb{R}^{d\times p}$, which can be obtained using the chain rule

$$\frac{\partial \boldsymbol{\phi}_1^j(\mathbf{x})}{\partial \boldsymbol{\theta}} = \frac{\partial \boldsymbol{\phi}_1^j}{\partial \mathbf{x}'}\frac{\partial \mathbf{x}'}{\partial \boldsymbol{\theta}} \text{ for } j = 1, \dots, d, \qquad (3)$$

where $\partial \boldsymbol{\phi}_1^j / \partial \mathbf{x}'$ is estimated stochastically through x- and y-finite difference filters on $\boldsymbol{\phi}_1^j$, and $\partial \mathbf{x}' / \partial \boldsymbol{\theta}$ is obtained deterministically from the warp function. The multi-channel $d \times p$ Jacobian matrix can then be formed as

$$\mathbf{J}_{\boldsymbol{\phi}}(\mathbf{x};\boldsymbol{\theta}) = \frac{\partial \boldsymbol{\phi}_1(\mathbf{x})}{\partial \boldsymbol{\theta}} = \begin{bmatrix} \frac{\partial \boldsymbol{\phi}_1^1(\mathbf{x})}{\partial \boldsymbol{\theta}} & \dots & \frac{\partial \boldsymbol{\phi}_1^d(\mathbf{x})}{\partial \boldsymbol{\theta}} \end{bmatrix}^\top. \qquad (4)$$

Using this multi-channel linearization, extensions and variations of the LK algorithm can be used with different multi-channel descriptors [1, 6]. In the next section, we demonstrate the application of the multi-channel LK algorithm to matching binary descriptors.

3. Lucas-Kanade with Binary Descriptors

In this work, we consider the simplest form of binary descriptors: The Census Transform (CT) [43]. The CT

8	12	200
56	42	55
128	16	11

(a)

8<42	12<42	200<42
56<42		55<42
128<42	16<42	11<42

(b)

1	1	0
0		0
0	1	1

(c)

Figure 3: Illustrating of the Census Transform. In Fig. 3a the center pixel is compared to its neighbors as shown in Fig. 3b. The descriptor is obtained by combining the results of each comparison in Fig. 3c into a single scalar.

Figure 4: An example of the CT evaluated on a 3×3 neighborhood, which results in an 8-channel Bit-Planes descriptor.

is based on the predicate of pixel comparisons in a small neighborhood as illustrated in Fig. 3 and is commonly used in robust optical flow [17] and stereo [41, 20]. The same idea was independently developed under the name Local Binary Patterns (LBP) [30]. We will use the CT term throughout as LBP is commonly associated with a non-binary histogrammed version [31]. By definition, the CT is invariant to monotonic changes in intensity. Its computation can be performed efficiently as it relies on comparison operators in a small 3×3 neighborhood. Since there are only eight comparisons per pixel, the CT signature is composed of an 8-bit pattern, which is commonly stored in a compact form as a single channel image as illustrated in Fig. 4. A byte in the CT image at pixel location $\mathbf{x}$ is given by

$$\phi(\mathbf{x}) = \sum_{i=1}^{8} 2^{i-1} \big[\mathbf{I}(\mathbf{x}) \bowtie \mathbf{I}(\mathbf{x} + \Delta\mathbf{x}_i)\big], \qquad (5)$$

where $\{\Delta\mathbf{x}_i\}_{i=1}^{8}$ is the set of the eight relative coordinate displacements possible within a 3×3 neighborhood around the center pixel location $\mathbf{x}$. Other neighborhood sizes and sampling locations can be used, but we found a 3×3 region to perform well. The operator $\bowtie \in \{>, \geq, <, \leq\}$ is a comparison, and the bracket denotes the indicator function.

Due to the compactness of a single-channel descriptor, one may be tempted to use it directly for tracking in lieu of the original image intensities. Nonetheless, the use of the single-channel representation produces biased estimates due to the dependence on arbitrary pixel ordering as we will demonstrate in the sequel. The alternative is to separate the descriptor into multiple channels composed of bits to produce that we call Bit-Planes as visualized in Fig. 4.

3.1. The Bit-Planes descriptor

When matching binary descriptors, it is a common practice to employ the Hamming distance. This is important because the Hamming distance is invariant to the ordering of pixel comparisons within the neighborhood used to compute the descriptor. In contrast, the sum or squared distances (SSD) lacks this desirable property and is dependent on the ordering specified by $\{\Delta\mathbf{x}_i\}_{i=1}^{8}$. This becomes problematic when employing dense binary descriptors within the LK framework due to its inherent dependence on the SSD. To make dense binary descriptors compatible with LK we propose the *Bit-Planes* descriptor given by:

$$\boldsymbol{\phi}(\mathbf{x}) = \begin{bmatrix} \mathbf{I}(\mathbf{x}) \bowtie \mathbf{I}(\mathbf{x} + \Delta\mathbf{x}_1) \\ \vdots \\ \mathbf{I}(\mathbf{x}) \bowtie \mathbf{I}(\mathbf{x} + \Delta\mathbf{x}_8) \end{bmatrix} \in \mathbb{R}^{8\times1} . \qquad (6)$$

For each pixel coordinate $\mathbf{x}$ in the image, this descriptor produces an 8-channel binary-valued vector. Notably, using the SSD with the multi-channel representation in Eq. (6) between two Bit-Planes descriptors is equivalent to the Hamming distance between the single-channel CT images. Specifically, the ordering of the pixel comparisons within the 3×3 neighborhood of the Bit-Planes descriptor has no effect on the SSD.

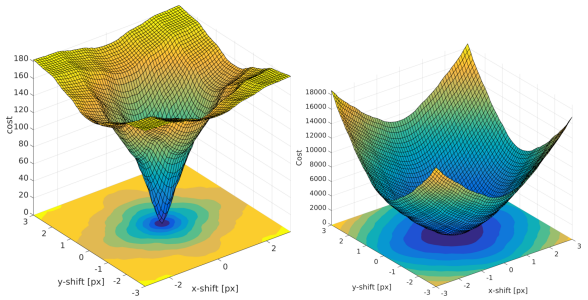
The Hamming distance is defined as the sum of mismatched bits between two binary strings [18]. To illustrate the equivalence between the Hamming distance and the sum of squared errors using Bit-Planes we use an example composed of three bits. Let $\mathbf{a} = \{1, 0, 1\}$, and $\mathbf{b} = \{0, 1, 1\}$. The Hamming distance between $\mathbf{a}$ and $\mathbf{b}$ is 2 as the two bit strings differ at two locations. The sum of squared differences between $\mathbf{a}$ and $\mathbf{b}$ is given by $(1-0)^2 + (0-1)^2 + (1-1)^2$, which is identical to the Hamming distance.

4. Linearizing Bit-Planes

In this section we answer a number of important questions regarding the validity of the dense Bit-Planes descriptor for robust and efficient image alignment.

In order for the Bit-Planes descriptor to be effective within a multi-channel LK framework we need to ensure the existence of an approximate linear relationship between the Bit-Planes and geometric displacements. Inspecting a depiction of the descriptor in Fig. 4, one might be doubtful about the existence of such relationship as each channel of the descriptor is discontinuous. In addition, estimating stochastic gradients per binary channel seems strange as they can take on only a handful of possibilities

The news is not all gloomy. In Fig. 5b we see the SSD cost surface between a patch within a natural image and shifted versions of itself in the x- and y- directions averaged over a subset of natural images. As expected, we observe the quasi-convex cost surface for raw pixel intensities. The shape of the cost surface is important to the effectiveness of the LK algorithm — as the LK objective relies on a graceful reduction of the SSD cost as a function of geometric displacement. Interestingly, when inspecting Fig. 5a we see a similar quasi-convex cost surface, albeit not as wide, which indicates that Bit-Planes have similar properties to raw pixel intensities when using the SSD as a measure of dissimilarity. The disadvantage, however, is a narrower basin of convergence in comparison to using raw intensities.



(a) Bit-Planes. (b) Raw intensity.

Figure 5: Cost surface of our Bit-Planes descriptor Fig. 5a computed over a subset of natural images [42] in comparison to the SSD over raw intensity Fig. 5b. Both surfaces are suitable for LK.

4.1. Quality of linearization

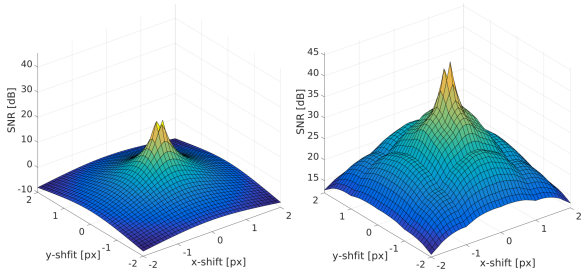
Consider a translational displacement warp $\Delta\boldsymbol{\theta} \in \mathbb{R}^2$ where we attempt to linearly predict an image representation $\mathbf{R}$ (raw pixels $\mathbf{I}$, or Bit-Planes $\boldsymbol{\phi}$) in the x- and y- directions, $\mathbf{R}(\mathbf{x}(\mathbf{0})) + \frac{\partial \mathbf{R}(\mathbf{0})}{\partial \boldsymbol{\theta}} \Delta\boldsymbol{\theta} \approx \mathbf{R}(\mathbf{x}(\Delta\boldsymbol{\theta}))$. The error of this linear approximation is given by

$$\boldsymbol{\epsilon}(\Delta\boldsymbol{\theta}) = \textstyle\sum_{\mathbf{x}\in\Omega} \| \mathbf{R}(\mathbf{x}(\mathbf{0})) + \frac{\partial \mathbf{R}(\mathbf{0})}{\partial \boldsymbol{\theta}} \Delta\boldsymbol{\theta} - \mathbf{R}(\mathbf{x}(\Delta\boldsymbol{\theta})) \|_2^2 \,, \quad (7)$$

and its signal-to-noise-ratio (SNR) can be computed using

$$\text{SNR}(\Delta\boldsymbol{\theta}) = 10 \cdot \left(\log \textstyle\sum_{\mathbf{x}\in\Omega} \| \mathbf{R}(\mathbf{x}(\mathbf{0})) \|_2^2 - \log \boldsymbol{\epsilon}(\Delta\boldsymbol{\theta}) \right) . \quad (8)$$

In Fig. 6 we show the SNR of the linearized objective as a function of increasing translational shifts from the true minima for both raw intensities, and Bit-Planes. The experiments were carried out similarly through the use of a subset of natural images and aggregated to form the results in Fig. 6. As expected, the SNR when using binary features is lower than using raw intensities due to the additional quantization when using binary data. However, it seems that — at least qualitatively — Bit-Planes gradient estimates provide a good local linear approximation of the objective.



(a) Bit-Planes. (b) Raw intensity.

Figure 6: Assessment of the linearization properties of the Bit-Planes descriptor in terms of the signal-to-noise-ratio (SNR) as a function of translational displacement. Even though raw pixels enjoy a higher SNR, using Bit-Planes offers a sufficient approximation for a gradient-based optimization framework.

Table 1: Planar template tracking runtime until convergence in frames per second on a single core Intel i7-2460M @ 2.8 Ghz.

	Template area			
	75×57	150×115	300×230	640×460
Intensity	650	360	140	45
Bit-Planes	460	170	90	35

5. Experiments

5.1. Comparison with LK using the CT

Employing Bit-Planes requires the alignment of eight separate channels as opposed to a single channel when working with raw intensities. In Section 3 we discussed the problems of using the CT within the LK framework. In particular, the representation is inherently sensitive to the ordering of pixel comparisons when using a SSD measure of dissimilarity. Using the CT within a LK framework as been reported to perform well [40, 17] given small displacements. However, under moderate displacements the use of the CT in LK introduces biases due to choices of the binary test and neighborhood ordering. In Fig. 8 we show the effect of differing binary comparison operators $\bowtie \in \{>, \geq, <, \leq\}$ compared to our proposed Bit-Planes descriptor. Our adaptation of the CT is unaffected by the ordering. In our experiments we noticed indistinguishable differences in performance between binary comparison operators when employing the Bit-Planes descriptor. As a result, we chose to use the $>$ operator for the rest of our experiments.

5.2. Real-time template tracking

We evaluate the performance of Bit-Planes for a template tracking problems using the benchmark dataset collected by

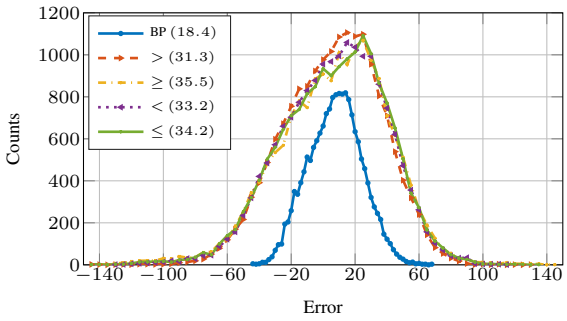


Figure 7: Histogram of intensity errors when using our Bit-Planes (BP) vs. a single-channel Census Transform (CT) with different comparison operators The RMSE is shown in parenthesis.

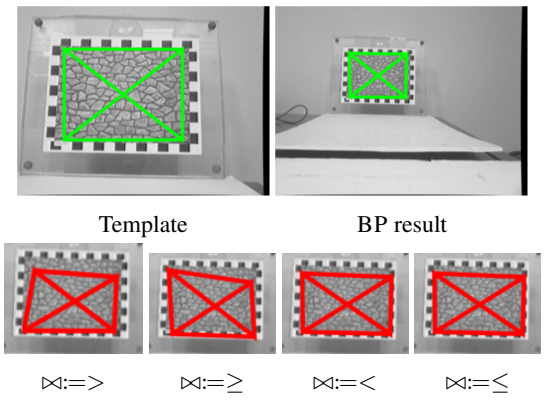


Figure 8: Drift when using the CT vs. Bit-Planes. The bottom row shows the result of template tracking using a single-channel CT. The images are magnified for better visualization (compare with BP result). Best viewed in color.

Gauglitz *et al*. [16]. An example of the dataset is shown in Fig. 10. Our plane tracker estimates an 8DOF homography using the IC algorithm [3]. The template is extracted from the first frame in each sequence and is kept fixed throughout as we are interested in tracking robustness overtime. To improve convergence we use a 3-level pyramid and initialize the tracker for subsequent frames using the most recent estimate. We use Gauss-Newton as the optimization algorithm, without robust weighting, and with a maximum of 100 iterations. Tracking terminates early if the relative change in the estimated parameters drops below 1×10^{-6}, or the relative change in the cost function drops below 1×10^{-5}. For small motions, the tracker typically converges in less than 10 iterations using Bit-Planes, or raw intensities. Our implementation runs faster than real time as shown in Table 1. The efficiency is achieved by utilizing SIMD instructions on the CPU, which allow us to process 16 pixels at once (or 32 pixels with AVX instructions). Additionally, the operations required to compute the descriptor are limited to bit shifts, ORs and ANDs, all of which can be performed with high throughput and low latency.

Table 2: Algorithms compared in this work. Number of parameters indicate the DOF of the state vector, which is 8 for a plane in additon to any photometric parameters. We use the authors' code for ECC and DIC.

Algorithm	# parameters	# channels
BP (ours)	8	8
ECC [14]	8	1
DIC-1 [4]	10	1
DIC-2 [4]	20	3
DF [10]	8	5
GC [7]	8	3
GM	8	2
LK	8	1

We compare the performance of our algorithm against a variety of template tracking methods summarized in Table 2. The algorithms are: the Enhanced Correlation Coefficient **ECC** [14], which serves as an example of an intrinsically robust cost function that is invariant to affine illumination change. The Dual Inverse Compositional (DIC) algorithm [4], which severs as an example of algorithms that estimate illumination parameters alongside the motion. We use two variations of the DIC: (i) the gain+bias model on grayscale images denoted by **DIC-1**, and (ii) using a full affine lighting model the makes use of RGB data denoted by **DIC-2**. We also compare the performance against a recently published descriptor-based method [10] called Descriptor Fields **DF**. Finally, we include baseline results from raw intensity **LK**, improved LK with the Gradient Constraint **GC** [7], and alignment with the Gradient Magnitude **GM**.

We report two quantities in the evaluation. The first is the percentage of successfully tracked frames. A frame is successfully tracked if the overlap between the estimate and the ground truth is greater than 90%. The overlap is computed as $o = (A \cap B)/(A \cup B)$, where A is the warped image given each algorithm's estimate, and B is the warped image given the ground truth. Second, since we are also interested in subpixel accuracy we show the mean percentage of overlap across all frames given by $m = {}^1\!/_n \sum_{i=1}^{n} o_i$, where n is the number of frames in each sequence.

Results are compared for three types of geometric and photometric variations suitable for LK-based alignment. First is an **out of plane rotation**, which induces perspective change as shown in Fig. 10b. Second, is **dynamic lighting change** where the image is stationary but illuminated with nonlinearly varying light source. Finally, a **static lighting change**, where illumination change is sudden.

Our evaluation results are shown in Table 3 and in Fig. 9. The top performing methods are baed on a descriptor constancy assumption, namely: BP and DF. However, BP is

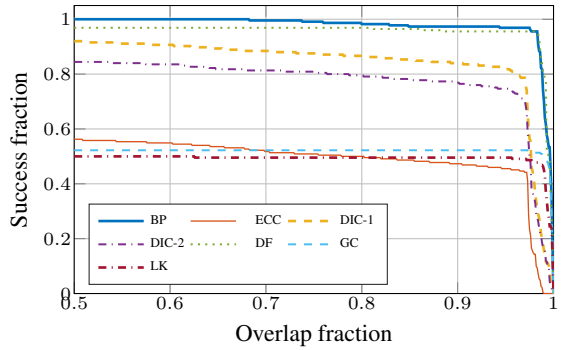
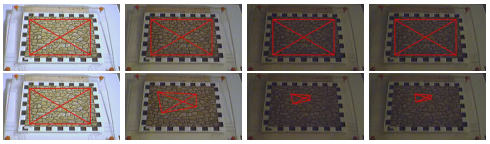
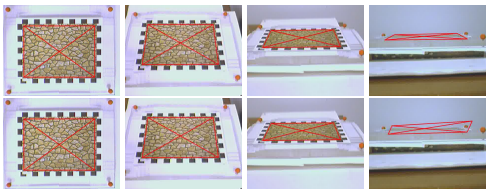


Figure 9: Fraction of successfully tracked frames as function of the overlap area given the ground truth. Bit-planes and DF perform better than other methods. However, in Table 3 we see that Bit-Planes' performance is better with challenging sequences.



(a) Lighting change.



(b) Out-of-plane rotation.

Figure 10: Tracking results using the Bricks dataset [16]. The top row of each figure shows the performance of Bit-Planes, while the bottom row shows classical intensity-based LK.

more efficient and it performed significantly better for the out of plane rotation data. In fact, all tested algorithms, except BP, performed poorly with this data. Algorithms that use a robust function (ECC) and ones that estimate illumination (DIC) performed well, but fell behind in comparison to descriptor constancy and even the gradient constraint.

5.3. Robustness to specular reflections

We use the "book" dataset [38] to illustrate robustness to severe specular reflections, where more than 50% of the template is oversaturated by specular reflections. The dataset does not contain groundtruth, so qualitative results are shown in Fig. 12 in comparison to other methods. Refereeing to Fig. 12, our approach maintains robust and accurate tracking under challenging specular reflections.

5.4. Results on mobile devices

We further evaluate the work on high frame rate data (Slo-mo) using two smart mobile devices: the iPad Air 2 and the iPhone 5s. In addition to compression artifacts, we made the data more challenging by flicking the lights multiples times during acquisition to cause sudden lighting change and low illumination. The videos are recorded with unsteady hands causing further motion blur. An example of the videos is shown in Fig. 11 featuring an ambiguously textured object in Fig. 11a, normal levels of texture in Fig. 11b as well as higher amount of texture in Fig. 11c. The first image starting from the left in Fig. 11 shows the selected template, which we hold fixed throughout tracking. The total number of frames from the videos combined is 6447.

We compare the performance of dense tracking using Bit-Planes with the RANSAC-based track by detection using two types of binary descriptors, ORB [35] and BRISK [25]. In terms of efficiency, even though our mobile device implementation does not make use of NEON instructions or the GPU, we outperform OpenCV3's optimized implementations of ORB and BRISK by a substantial margin. More importantly, our approach is more robust. Feature-based tracking failed on $\approx 15\%$ of the frames due to either the inability to detect features under low light, or failure due to imprecise correspondences under motion blur.

Perhaps more interestingly, Bit-Planes is able to maintain performance with smaller image resolution. In fact, tracking speed more than doubles when reducing the template size by half. However, this is not the case with sparse features as memory overhead depends on the number of extracted keypoints, which we kept fixed at 512. It is possible to improve the tracking speed of ORB and BRISK by reducing the number of extracted keypoints. However, lowering the number of keypoints must be done carefully as not to compromise the robustness of the system. We note that the ability to work with lower resolution is important on mobile devices to lower power consumption.

Finally, we note that while dense Bit-Planes tracking produces faster and more accurate results, its main limitation is the inability to recover if the template is lost due to occlusions, significant drift or large motions. In such cases, track by detection can be of immense value to re-initialize LK-based methods if needed.

6. Conclusions

In this work, we presented an algorithm for robust tracking in low and under sudden and arbitrary changes in illumination. The proposed algorithm is a novel adaption of binary feature descriptor to a multi-channel Lucas and Kanade algorithm. Under our formulation, we demonstrated the equivalence of the Hamming distance to the sum of squared difference. Hence, the proposed tracking ap-

Table 3: Template tracking evaluation [16]. We show the percentage of successfully tracked frames. In parenthesis we show the average percentage of overlap for all successfully tracked frames. The available textures are: `br` (bricks), `bu` (building), `mi` (mission), `pa` (paris), `su` (sunset), and `wd` (wood).

	br	bu	mi	pa	su	wd
			Out of Plane Rotation			
BP	**100.0** (99.38)	**100.0** (99.51)	**87.50** (99.38)	**97.92** (99.26)	**79.17** (99.57)	**93.75** (99.30)
ECC	25.00 (96.16)	33.33 (95.85)	25.00 (95.99)	33.33 (96.65)	20.83 (95.52)	18.75 (95.14)
DIC-1	25.00 (96.20)	33.33 (95.83)	25.00 (95.98)	33.33 (96.73)	20.83 (95.95)	18.75 (95.46)
DIC-2	25.00 (96.22)	35.42 (95.56)	25.00 (95.51)	35.42 (96.42)	25.00 (96.22)	18.75 (95.06)
DF	91.67 (99.51)	93.75 (99.44)	79.17 (99.70)	85.42 (99.75)	70.83 (99.60)	83.33 (99.51)
GC	**100.0** (99.24)	95.83 (99.66)	87.50 (99.52)	93.75 (99.51)	62.50 (98.88)	91.67 (99.34)
GM	62.50 (99.86)	83.33 (99.62)	77.08 (99.72)	77.08 (99.81)	58.33 (99.71)	62.50 (99.66)
LK	93.75 (99.68)	91.67 (99.70)	83.33 (99.32)	91.67 (99.63)	37.50 (97.64)	66.67 (99.63)
			Dynamic Lighting Change			
BP	**100.0** (98.97)	**100.0** (99.08)	**100.0** (99.13)	**100.0** (98.91)	**100.0** (98.98)	**100.0** (99.02)
ECC	16.33 (98.03)	19.39 (99.00)	**100.0** (98.64)	**100.0** (98.69)	**100.0** (97.30)	67.35 (98.55)
DIC-1	**100.0** (98.40)	**100.0** (99.04)	**100.0** (98.77)	**100.0** (98.60)	86.87 (96.02)	20.41 (95.36)
DIC-2	**100.0** (98.39)	**100.0** (98.85)	**100.0** (98.61)	**100.0** (98.58)	85.86 (96.42)	26.53 (97.73)
DF	**100.0** (99.30)	**100.0** (99.08)	**100.0** (98.35)	**100.0** (98.87)	20.41 (99.36)	68.37 (99.02)
GC	17.35 (99.87)	**100.0** (99.50)	22.45 (99.84)	18.37 (99.88)	12.24 (99.72)	17.35 (99.84)
GM	17.35 (98.99)	19.39 (99.23)	23.47 (99.10)	19.39 (99.08)	0.00 (0.00)	0.00 (0.00)
LK	13.27 (99.34)	31.63 (98.26)	18.37 (98.82)	18.37 (99.32)	12.24 (99.16)	16.33 (98.96)
			Static lighting change			
BP	**100.0** (99.76)	**100.0** (99.85)	**100.0** (99.61)	**100.0** (99.85)	**100.0** (99.63)	**100.0** (99.76)
ECC	**100.0** (97.33)	**100.0** (97.67)	**100.0** (97.75)	**100.0** (97.41)	**100.0** (96.79)	**100.0** (97.55)
DIC-1	**100.0** (97.70)	**100.0** (97.77)	**100.0** (97.80)	**100.0** (97.20)	98.72 (96.58)	89.74 (96.19)
DIC-2	**100.0** (97.58)	79.49 (97.59)	**100.0** (97.07)	**100.0** (97.13)	89.74 (95.75)	79.49 (96.38)
DF	**100.0** (99.68)	**100.0** (99.51)	76.92 (99.71)	**100.0** (99.77)	74.36 (99.70)	**100.0** (99.83)
GC	74.36 (99.73)	74.36 (99.84)	48.72 (99.97)	74.36 (99.76)	48.72 (99.74)	51.28 (99.88)
GM	48.72 (99.88)	74.36 (99.75)	74.36 (99.66)	74.36 (99.81)	48.72 (99.76)	48.72 (99.83)
LK	48.72 (99.80)	74.36 (99.67)	48.72 (99.95)	48.72 (99.93)	48.72 (99.40)	48.72 (99.94)

Table 4: Template tracking running time on ARM architecture using a single CPU core in frames per second (FPS). The bottleneck for Bit-Planes is image resizing and warping, which could be mitigated using the GPU. Results are averaged over three videos of challenging data totalling 6446 frames.

	iPad Air 2			iPhone 5s		
template size	BP	ORB	BRISK	BP	ORB	BRISK
70×55	**123**	N/A	N/A	**50**	N/A	N/A
150×115	**48**	15	15	**22**	13	13
311×230	**17**	12	14	**10**	8	11

proach maintains the photometric invariance properties enjoyed by binary descriptors. In addition to robustness, we obtained precise subpixel localization of binary descriptors at speeds faster than real-time on laptops and mobile devices.

References

[1] E. Antonakos, J. Alabort-i Medina, G. Tzimiropoulos, and S. Zafeiriou. Feature-Based Lucas-Kanade and Active Appearance Models. *Image Processing, IEEE Transactions on*, 24(9):2617–2632, Sept 2015. 2

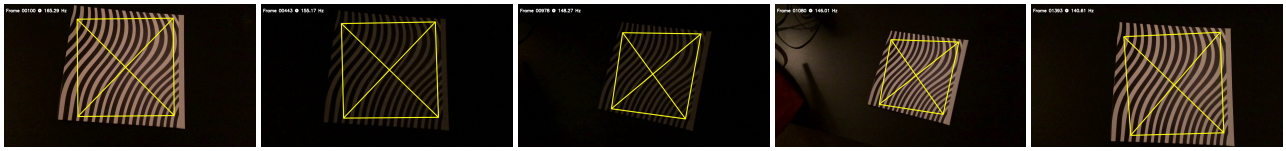
[2] S. Baker, R. Gross, and I. Matthews. Lucas-kanade 20 years on: A unifying framework: Part 3. Technical Report CMU-RI-TR-03-35, Robotics Institute, Pittsburgh, PA, 2003. 1

[3] S. Baker and I. Matthews. Lucas-kanade 20 years on: A unifying framework. *International Journal of Computer Vision*, 56(3):221–255, 2004. 2, 5

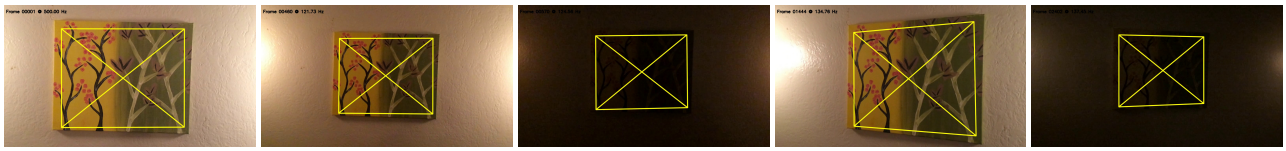
[4] A. Bartoli. Groupwise geometric and photometric direct image registration. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 30(12):2098–2108, 2008. 1, 5

[5] E. Bostanci. Is Hamming distance only way for matching binary image feature descriptors? *Electronics Letters*, 50(11):806–808, May 2014. 2

[6] H. Bristow and S. Lucey. In Defense of Gradient-Based Alignment on Densely Sampled Sparse Features. In *Dense correspondences in computer vision*. Springer, 2014. 2



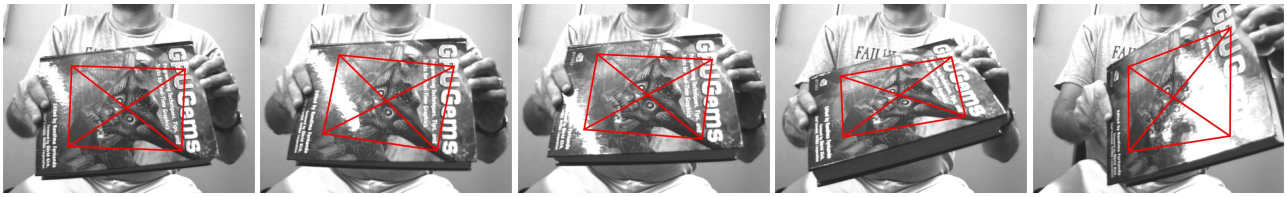
(a) Sudden lighting change and ambiguous texture.



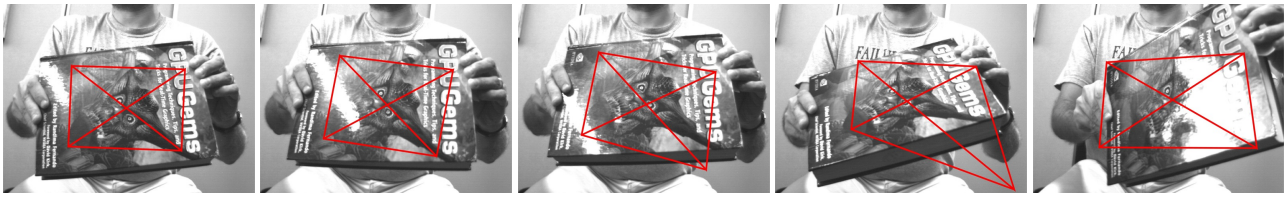
(b) Sudden lighting change and perspective distortion with medium texture.

(c) Sudden lighting change and motion blur with high texture.

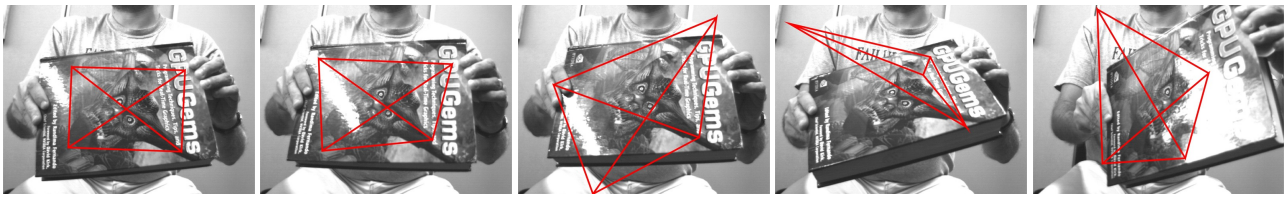
Figure 11: High frame rate data at 120 Hz captured using an iPhone 5s. Dataset contains different textures under sudden lighting change, low lighting, and motion blur. Video demonstration of the approach is available in the supplementary materials. The datasets are available at www.cs.cmu.edu/~halismai/bitplanes



(a) Bit-Planes.



(b) Descriptor Fields [10].



(c) Gradient Constraint

Figure 12: Illustration of robustness to specular reflections in comparision to other tracking algorithms using the "book" dataset [38]. Video demonstration is available in the supplementary materials.

[7] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert. High Accuracy Optical Flow Estimation Based on a Theory for Warping. In *ECCV*, volume 3024. 2004. 5

[8] T. Brox and J. Malik. Large displacement optical flow: descriptor matching in variational motion estimation. *IEEE transactions on pattern analysis and machine intelligence*, 33(3):500–513, 2011. 2

[9] T. Cootes, G. Edwards, and C. Taylor. Active appearance

models. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 23(6):681–685, Jun 2001. 2

[10] A. Crivellaro and V. Lepetit. Robust 3D Tracking with Descriptor Fields. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014. 2, 5, 8

[11] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition (CVPR). IEEE Computer Society Conference on*, volume 1, pages 886–893 vol. 1, June 2005. 2

[12] A. Dame and E. Marchand. Accurate real-time tracking using mutual information. In *Mixed and Augmented Reality (ISMAR), 2010 9th IEEE International Symposium on*, pages 47–56, Oct 2010. 1

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

[14] G. D. Evangelidis and E. Z. Psarakis. Parametric image alignment using enhanced correlation coefficient maximization. *PAMI*, 30(10), 2008. 1, 5

[15] J. Figat, T. Kornuta, and W. Kasprzak. Performance Evaluation of Binary Descriptors of Local Features. In *Computer Vision and Graphics*, pages 187–194. Springer, 2014. 2

[16] 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. 5, 6, 7

[17] D. Hafner, O. Demetz, and J. Weickert. Why Is the Census Transform Good for Robust Optic Flow Computation? In *Scale Space and Variational Methods in Computer Vision*, volume 7893. Springer Berlin Heidelberg, 2013. 2, 3, 4

[18] R. W. Hamming. Error detecting and error correcting codes. *Bell System technical journal*, 29(2):147–160, 1950. 3

[19] J. Heinly, E. Dunn, and J.-M. Frahm. Comparative evaluation of binary features. In *Computer Vision–ECCV 2012*, pages 759–773. Springer, 2012. 2

[20] H. Hirschmuller and D. Scharstein. Evaluation of stereo matching costs on images with radiometric differences. *IEEE transactions on pattern analysis and machine intelligence*, 31(9):1582–1599, 2009. 3

[21] B. K. Horn and B. G. Schunck. Determining optical flow. *Artificial intelligence*, 17(1):185–203, 1981. 1

[22] M. Irani and P. Anandan. Robust multi-sensor image alignment. In *Computer Vision, 1998. Sixth International Conference on*, pages 959–966, Jan 1998. 1

[23] M. Irani and P. Anandan. About Direct Methods. In *Vision Algorithms: Theory and Practice*, pages 267–277. Springer Berlin Heidelberg, 2000. 1

[24] J.-S. Kim, M. Hwangbo, and T. Kanade. Realtime affine-photometric KLT feature tracker on gpu in cuda framework. In *ICCV Workshops, IEEE 12th International Conference on*, pages 886–893. IEEE, 2009. 1

[25] S. Leutenegger, M. Chli, and R. Siegwart. BRISK: Binary Robust invariant scalable keypoints. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 2548–2555, Nov 2011. 6

[26] D. G. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004. 2

[27] B. D. Lucas and T. Kanade. An Iterative Image Registration Technique with an Application to Stereo Vision (DARPA). In *Proc. of the 1981 DARPA Image Understanding Workshop*, pages 121–130, April 1981. 1

[28] T. Mller, C. Rabe, J. Rannacher, U. Franke, and R. Mester. Illumination-Robust Dense Optical Flow Using Census Signatures. In *Pattern Recognition*. 2011. 2

[29] J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer, New York, 2nd edition, 2006. 1

[30] T. Ojala, M. Pietikäinen, and D. Harwood. A comparative study of texture measures with classification based on featured distributions. *Pattern Recognition*, 29:51–59, 1996. 3

[31] T. Ojala and M. Pietikinen. Unsupervised texture segmentation using feature distributions. In *Image Analysis and Processing*, volume 1310, pages 311–318. 1997. 3

[32] G. Panin and A. Knoll. Mutual Information-Based 3D Object Tracking. *International Journal of Computer Vision*, 78(1):107–118, 2008. 1

[33] S. Ren, X. Cao, Y. Wei, and J. Sun. Face Alignment at 3000 FPS via Regressing Local Binary Features. In *CVPR*, pages 1685–1692, June 2014. 2

[34] R. Richa and H. Gregory. Robust Similarity Measures for Gradient-based Direct Visual Tracking. Technical report, CIRL, 2012. 1

[35] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski. ORB: An efficient alternative to SIFT or SURF. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 2564–2571, Nov 2011. 6

[36] L. Sevilla-Lara and E. Learned-Miller. Distribution fields for tracking. In *Computer Vision and Pattern Recognition (CVPR), IEEE Conference on*, 2012. 2

[37] L. Sevilla-Lara, D. Sun, E. G. Learned-Miller, and M. J. Black. *Optical Flow Estimation with Channel Constancy*, pages 423–438. Springer International Publishing, 2014. 2

[38] G. Silveira and E. Malis. Real-time Visual Tracking under Arbitrary Illumination Changes. In *Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on*, pages 1–6, June 2007. 6, 8

[39] F. Stein. Efficient Computation of Optical Flow Using the Census Transform. In *Pattern Recognition*, volume 3175 of *Lecture Notes in Computer Science*, pages 79–86. Springer Berlin Heidelberg, 2004. 2

[40] C. Vogel, S. Roth, and K. Schindler. An Evaluation of Data Costs for Optical Flow. In J. Weickert, M. Hein, and B. Schiele, editors, *Pattern Recognition*, Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2013. 2, 4

[41] J. I. Woodfill, G. Gordon, and R. Buck. Tyzx deepsea high speed stereo vision system. In *Computer Vision and Pattern Recognition Workshop, 2004. CVPRW'04. Conference on*, pages 41–41. IEEE, 2004. 3

[42] J. Xiao, J. Hays, K. Ehinger, A. Oliva, A. Torralba, et al. Sun database: Large-scale scene recognition from abbey to zoo. In *Computer vision and pattern recognition (CVPR), 2010 IEEE conference on*, pages 3485–3492. IEEE, 2010. 4

[43] R. Zabih and J. Woodfill. Non-parametric local transforms for computing visual correspondence. In *Computer Vision - ECCV'94*, pages 151–158. Springer, 1994. 1, 2

[44] C. Zach, D. Gallup, and J.-M. Frahm. Fast gain-adaptive klt tracking on the gpu. In *Computer Vision and Pattern Recognition Workshops, 2008. CVPRW'08. IEEE Computer Society Conference on*, pages 1–7. IEEE, 2008. 1