# Efficient Computation of Optical Flow Using the Census Transform

Fridtjof Stein

DaimlerChrysler AG,
Research and Technology,
D-70546 Stuttgart, Germany
Fridtjof.Stein@DaimlerChrysler.com

**Abstract.** This paper presents an approach for the estimation of visual motion over an image sequence in real-time. A new algorithm is proposed which solves the correspondence problem between two images in a very efficient way. The method uses the Census Transform as the representation of small image patches. These primitives are matched using a table based indexing scheme. We demonstrate the robustness of this technique on real-world image sequences of a road scenario captured from a vehicle based on-board camera. We focus on the computation of the optical flow. Our method runs in real-time on general purpose platforms and handles large displacements.

## 1 Introduction

Recovering motion information from a visual input is a strong visual cue for understanding structure and three-dimensional motion. Visual motion allows us to compute properties of the observed three-dimensional world without the requirement of extensive knowledge about it.

In this paper we propose a strategy to efficiently determine the correspondences between consecutive image frames. The goal is to retrieve a set of promising image-to-image correspondence hypotheses. These correspondences are the basis for the computation of the optical flow. First we select a robust and descriptive primitive type. Then we match all primitives in one image with all the primitives in the consecutive image with a table based structural indexing scheme. Using structure as a means of correspondence search yields a matching method without search area limits.

This requires a computational complexity of $O(n)$ with $n$ the number of pixels in the image. It is obvious that the number of matches has a complexity of $O(n^2)$ which is an intractable high number for a typical image with $n = \text{const} * 10^5$ pixels.

We describe a pruning technique based on *discriminative power* to reduce this complexity to $O(C * n)$. $C$ is a small constant value. Discriminative power in this context is the descriptiveness of a primitive. The discriminative power of a primitive is inverse proportional to its occurrence frequency in an image. Temporal constraints further reduce the number of correspondence hypotheses to the resulting optical flow.

The structure of this paper is as follows: In the next section we give a brief overview of some previous work. In section 3 we discuss the feature type which we use: the Census Transform. The algorithm is presented in section 4. Section 5 explains the involved parameters. Some results illustrate the performance in section 6.

## 2 Related Work

In recent years a large amount of different algorithms for the computation of optical flow was developed. A good overview was given by Barron et al. [1] and by Cédras et al. [2]. Barron et al. classify the optical flow techniques in four classes: differential methods, region-based matching, energy-based techniques, and phase-based approaches. All of these methods have to handle trade-offs between computational efficiency, the maximum length of the flow, the accuracy of the measurements, and the density of the flow. In order to tackle the real-time constraint (see e.g. [3–6]), several strategies were followed:

- Strong limitations on the maximum allowed translation between frames increases efficiency even for correlation based approaches.
- Image pyramids or coarse sampling lower the computational burden. E.g. using only selective features, and tracking them over an image sequence requires little computational power.
- Some methods were implemented on dedicated hardware to achieve real-time performance (see e.g. [3, 6]).

Our method puts an emphasis on computational efficiency, and it allows a large span of displacement vector lengths. The accuracy is limited to pixel precision, and the density is texture driven. In areas with a lot of structural information the density is higher than in areas with little texture.

## 3 The Census Transformation

A very robust patch representation, the Census Operator, was introduced by Zabih and Woodfill [7]. It belongs to the class of non-parametric image transform-based matching approaches [8].

The Census Transform $R(P)$ is a non-linear transformation which maps a local neighborhood surrounding a pixel P to a binary string representing the set of neighboring pixels whose intensity is less than that of $P$. Each Census digit $\xi(P, P')$ is defined as

$$\xi(P, P') = \begin{matrix} 0 & P > P' \\ 1 & P \leq P' \end{matrix}$$

This is best demonstrated with an example. The left table shows the intensity values, the center table illustrates the Census values, and the right number is the corresponding clockwise unrolled *signature vector*.

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 124 | 74 | 32 | → | 1 | 1 | 0 | → | 11001111 | | |
| 124 | 64 | 18 | | 1 | x | 0 | | | | |
| 157 | 116 | 84 | | 1 | 1 | 1 | | | | |
| intensity values | | | | Census digits | | | | signature vector | | |

We extended the Census Transform by introducing the parameter $\varepsilon$ in order to represent “similar” pixel. This results in ternary signature vector digits.

$$\xi(P,P') = \begin{matrix} 0 & P - P' > \varepsilon \\ 1 & |P - P'| \leq \varepsilon \\ 2 & P' - P > \varepsilon \end{matrix}$$

$$\begin{array}{|c|c|c|} \hline 124 & 74 & 32 \\ \hline 124 & 64 & 18 \\ \hline 157 & 116 & 84 \\ \hline \end{array} \rightarrow \begin{array}{|c|c|c|} \hline 2 & 1 & 0 \\ \hline 2 & x & 0 \\ \hline 2 & 2 & 2 \\ \hline \end{array} \rightarrow 21002222$$

We decided to use the Census Transform as the basic primitive due to its robustness with respect to outliers, and its simplicity to compute. In addition the Census primitive is highly discriminative. A signature vector of length $c$ (its cardinality) represents $3^c$ different patches. This implicit discriminative power is the key to the proposed algorithm. The shape of the primitive, rectangular, circular, or star-like, is not significant.

While other approaches [7] use the Census Transform for correlation based matching we use it as an index into a table-based indexing scheme as described in the next section.

The correlation based approaches use the Hamming Distance for deciding whether two patches are similar. Our approach requires a Hamming Distance of zero. Therefore we lose a certain amount of "near matches" due to the binning effect. Beis and Lowe [9] discuss the issue of indexing in higher dimensions in their article. The exact analysis of this loss is part of our current research.

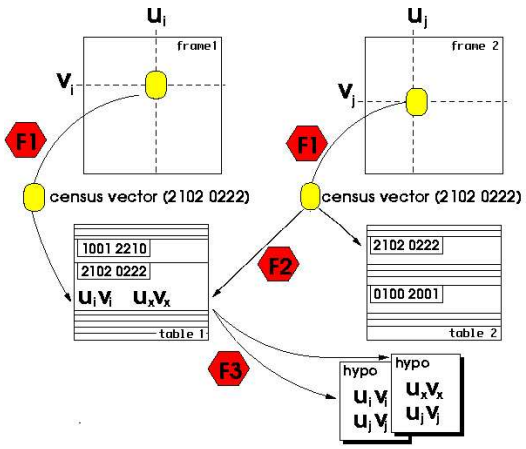


**Fig. 1.** The core algorithm with the resulting set of correspondence hypotheses.

## 4 The Algorithm

### 4.1 Finding Correspondences

All images are slightly low-pass filtered with a 3x3 mean filter before applying the algorithm below. All tables are implemented as hash-tables. The following steps are illustrated in Figure 1:

1. Scan image frame 1. Compute for every pixel $P_i^1 = (u_i^1, v_i^1)$ the signature $\xi(P_i^1)$.
2. The ternary $\xi(P_i^1)$ is interpreted as a decimal number and serves as the key to a table 1 in which the corresponding coordinate $(u_i^1, v_i^1)$ is stored.

3. Scan image frame 2. Compute for every pixel $P_j^2 = (u_j^2, v_j^2)$ the signature $\xi(P_j^2)$.
4. Look for every $\xi(P_j^2)$ in table 1 whether there are one or more entries with the same signature vector.
5. All the resulting $(u_i^1, v_i^1) \leftrightarrow (u_j^2, v_j^2)$ pairs represent correspondence hypotheses.
6. If a consecutive image (e.g. in optical flow) has to be analyzed create a table 2 from all the $\xi(P_j^2)$.

It is obvious that this procedure leads to a huge amount of correspondence hypotheses. E.g. a patch with uniform intensity values and a center pixel $U$ results in $\xi(U) = 1^c$. In our test images such uniform patches account for at least $10^4$ such patches. Therefore we have to analyze at least $10^8$ correspondence hypotheses.

The question is: How can we handle the explosion of the number of correspondence hypotheses?

In our work we use filters. They are labeled in Figure 1 with **F1** to **F3**.

**F1** Some patch patterns do not contribute to any meaningful correspondence computation. They are filtered out early. Typical patches are the above mentioned uniform patch, or all the patches which are related to the aperture problem. An example is depicted in Figure 2. F1 compares the retrieved signature vector with a list (called the F1-list) of candidates, and inhibits the further processing if found in this table. This list is learned on example sequences.

**F2** F2 introduces the parameter *max_discriminitive_power* ($= mdp$). F2 inhibits the correspondence hypothesis generation if a matching table entry has more than $mdp$ elements. Hypotheses are only generated if there are fewer elements. F2 filters out patches which were not yet stored in the F1-list.

**F3** The third filter uses illumination and geometric constraints to filter out unlikely hypotheses. Without loss of generality we typically allow an intensity change of the center pixel of 20% and we limit the displacement vector length to 70 pixel.

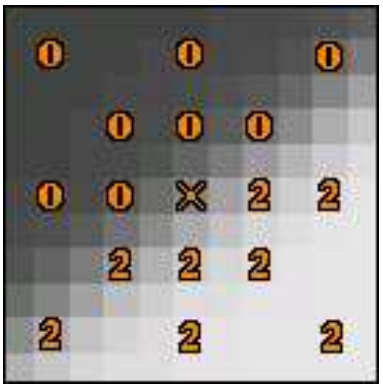


**Fig. 2.** Representation of an edge with the signature vector 0002 2220 0002 2220 and a frequency of occurrence of 1875 in Figure 6

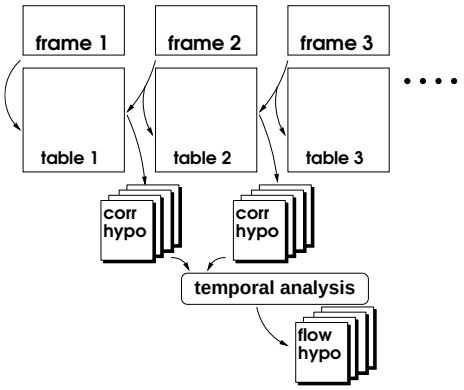


**Fig. 3.** Using temporal constraints.

### 4.2 Temporal Analysis

The algorithm in the previous section produces a large set of correspondence hypotheses. They are created based on *similarity*, not based on any flow constraints. Using the correspondence hypotheses of the two previous frames (Figure 3), and attaching a certain inertia to every displacement vector, we get a geometrical continuation constraint, illustrated in Figure 4.

Such an inertia constraint filters out all flow hypotheses which have no predecessor among the correspondence matches. A valid predecessor has

- a similar direction angle,
- it is located close to the actual flow hypothesis (with respect to a moderate catch radius $r$ as depicted in Figure 4),
- and its vector length has not changed too much.

Veenman et al. [10] address the temporal benefits extensively in their paper. It is important to note that our approach allows multiple associations as shown in Figure 5. We apply no heuristics to get a one-to-one correspondence. It is for the user to decide which interpretation serves him best.

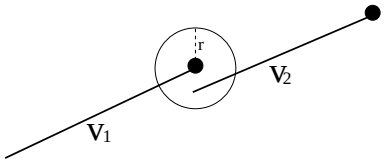


**Fig. 4.** Two consecutive correspondence vectors: $v_1$ originates from a correspondence between frame 1 and frame 2, $v_2$ originates from a match between frame 2 and frame 3.

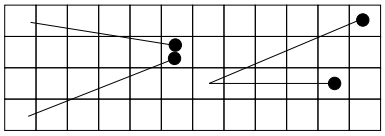

**Fig. 5.** Multiple interpretations have to be resolved later.

## 5 Parameters

In the last sections several parameters were introduced. Their effect on the overall performance is as follows:

**Census $\varepsilon$ :** This parameter was introduced in Section 3. $\varepsilon$ represents the *similarity* between Census features. If $\varepsilon = 0$, the signature vector becomes more sensitive to noise. If $\varepsilon = \text{grey}_{\text{max}}$, then $\xi(P_i) = 1^c$ for all pixel $P_i$. All patches have the same representation, and no discriminative power.
A typical value for 8 bit and 12 bit images is $\varepsilon = 16$.

**Census cardinality c:** The trade-off is discriminative power versus locality. Choosing a larger cardinality results in a patch representation with a higher discriminative power, but less locality. However, very local representation result in a high number of correspondence hypotheses.
For our road scenes we got the best results with large cardinalities (e.g. 20).

**Census sampling distance:** The example in Section 3 shows the design of a Census operator of cardinality 8 representing a neighborhood with a sampling distance of 1. However, we also looked at a sampling distance of 2 (skipping the adjacent neighbor pixel). The results improved dramatically. This supports an often ignored truth, that adjacent pixel values in natural images are not completely independent of each other [11].

**max_discriminitive_power:** This parameter represents the maximal allowed discriminitive power of the Census features. Setting it to a very high value results in a lot of potential correspondence interpretations. The constant $C$ which was mentioned in Section 1 becomes very large.
The parameter $max\_discriminitive\_power$ is dependent on the cardinality of the selected Census Operator. Typical values are
$max\_discriminitive\_power_{c=8} = 12$, and
$max\_discriminitive\_power_{c=20} = 3$ or 2.

**temporal analysis geometry constraints:** Application driven we use a maximum direction angle change 10 deg, and we allow a vector length change of 30 %. The radius $r$ is below 3 pixel.

**min_age:** The parameter $min\_age$ corresponds to the number of predecessors by which a flow vector is supported. We set this value typically to 1 or 2. See Figure 6 and 7 for an example.

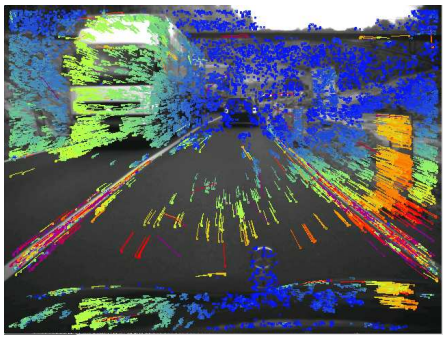

**Fig. 6.** A typical road scene. The displacement vectors are color coded with respect to their length. Warmer colors denote longer vectors. The displacement vectors in the lower right corner along the reflection post have a length of 40 pixels. The parameter $min\_age$ is set to 1.

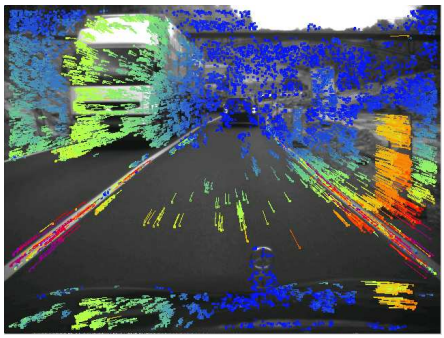

**Fig. 7.** The parameter $min\_age$ is set to 2. This results in fewer outliers, but also in a slightly reduced density.

## 6 Experimental Results

The following results on real images validate the approach. We applied to algorithm to a large set of image sequences recorded from a camera platform in a moving car. Figure 6

and Figure 7 show a frame from an image sequence of 12 bit images. in Figure 6 the parameter $min\_age$ is set to 1, in Figure 6 it is set to 2.

Notice, that even subtle distinctions on the road surface allow for the flow computation. The flow in the foreground is the flow on the hood of the car, functioning as a mirror of the scene.

Using a feature primitive type for matching always raises the question about its invariance properties. For the Census Transform the invariance with respect to translational motion is obvious. However, the signature vector is not implicitely invariant to rotational or scale transformations. We performed empirical tests on image sequences and observed the number of correspondences. We observe a graceful degradation with respect to rotation and scale. At an angle of $8$ deg, or alternatively at a scale of $0.8$ there are still half of the correspondences.

One of the contributions of our algorithm is its speed. The following times were measured on a 1.2 GHz Pentium III computer with our non-optimized program, implemented in C++. The scene in Figure 8 was processed. The image (frame) dimensions are 768 x 284. Every pixel was processed. The pixel depth is 8 bit.

| **parameter** | **value** |
|---|---|
| cardinality | 20 |
| $\varepsilon$ | 12 |
| min_age | 2 |
| max_discriminitive_power | 2 |

| **function** | **time [ms]** |
|---|---|
| smoothing | 5 |
| signature computation | 45 |
| table insertion | 30 |
| hypotheses generation | 22 |

Altogether 6577 correlation hypotheses were found. The 2513 flow hypotheses are shown in Figure 8.

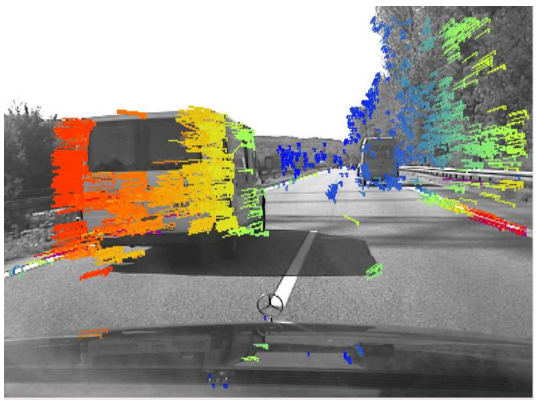

**Fig. 8.** An overtaking van. The original sequence consists of half frames. For aesthetical reasons the image is displayed vertically scaled. Notice that due to the filters **F1** and **F2** no displacements are computed along the shadow edge.

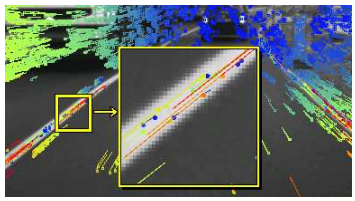

**Fig. 9.** Closeup from Figure 7: the aperture problem.

## 7 Future Research

As it can be seen in the closeup Figure 9, there are quite a few stable outliers along elongated edges. This is due to the imperfect nature of the filter F1. At the moment the adaption of the hard-coded F1-list is performed by collecting signature vectors with a high occurrence frequency. It is planned to automate this process.

At the moment we are investigating other "richer" representations of an image patch than the Census Transform. Here, richness is a synonym for discriminating power. We expect higher flow densities.

## 8 Summary and Conclusions

Though we present this paper in the context of optical flow, the approach serves as a general tool for pixel-precise image matching. It is applicable to other vision problems such as finding correspondences in stereo images. Due to the arithmetic-free nature of the algorithm, it has the potential of a straightforward implementation on an FPGA. For the Census operator itself this was already demonstrated in Woodfill et al. [12].

## References

1. J. L. Barron, D. J. Fleet, and S. S. Beauchemin, "Performance of optical flow techniques," *International Journal of Computer Vision*, vol. 12, no. 1, pp. 43–47, 1994.
2. C. Cedras and M. Shah, "Motion-based recognition: A survey," *IVC*, vol. 13, no. 2, pp. 129–155, March 1995.
3. P. C. Arribas and F. M. H. Macia, "FPGA Implementation of Camus Correlation Optical Flow Algorithm for real-time Images," .
4. R. Cutler and M. Turk, "View-based interpretation of real-time optical flow for gesture recognition," in *Third IEEE International Conference on Automatic Face and Gesture Recognition*, Nara, Japan, April 1998.
5. T.A. Camus and H.H. Bülthoff, "Real-time optical flow extended in time," Tech. Rep. 13, Tübingen, Germany, Feb 1995.
6. W. Enkelmann, V. Gengenbach, W. Krüger, S. Rössle, and W. Tölle, "Hindernisdetektion durch Real-Zeit-Auswertung von optischen Fluß-Vektoren," in *Autonome Mobile Systeme 1994*, P. Levi and T. Bräunl, Eds., pp. 285–295. Springer, Berlin, Heidelberg, 1994.
7. R. Zabih and J. Woodfill, "Non-parametric local transforms for computing visual correspondence," in *Proceedings of the Third European Conference on Computer Vision*, Stockholm, May 1994.
8. D. Bhat and S. Nayar, "Ordinal measures for visual correspondence," 1996, pp. 351–357.
9. Jeffrey S. Beis and David G. Lowe, "Indexing without invariants in 3d object recognition," *PAMI*, vol. 21, no. 10, pp. 1000–1015, 1999.
10. C.J. Veenman, M.J.T. Reinders, and E. Backer, "Establishing motion correspondence using extended temporal scope," vol. 145, no. 1-2, pp. 227–243, April 2003.
11. E. Trucco and A. Verri, *Introductory Techniques for 3-D Computer Vision*, Prentice Hall, 1998.
12. J. Woodfill and B. Von Herzen, "Real-time stereo vision on the parts reconfigurable computer," in *Proceedings IEEE Symposium on Field-Programmable Custom Computing Machines*, Napa, April 1997.
13. J. Shi and C. Tomasi, "Good features to track," in *IEEE Conference on Computer Vision and Pattern Recognition*, Seattle, 1994, pp. 592–600.