Dense Optic Flow with a Bayesian Occlusion Model

Kevin Koeser, Christian Perwass and Gerald Sommer

Institute of Computer Science and Applied Mathematics
Christian-Albrechts-University of Kiel
24098 Kiel, Germany
koeser@mip.informatik.uni-kiel.de
{chp, gs}@ks.informatik.uni-kiel.de

Abstract. This paper presents a dense optic flow algorithm based on quite simple local probability assumptions. Due to the explicit derivation of the correspondence concept in a probability-theoretical framework, occlusion probability evolves straight-forwardly from the model for each pixel. Initialized with a similarity measure based on single pixels, an iterated diffusion step propagates local information across the image, while occlusion probability is used to inhibit flow information transfer across depth discontinuities, which prevents flow smoothing at 3d object boundaries. The inhibition is thereby not artificially modelled by some heuristically chosen parameters, but arises directly from the Bayesian correspondence model. The algorithm structure can be interpreted as a recurrent neural network, where matched points have reached a stable state, while others (e.g. those in homogeneous areas) keep receiving information from regions more and more far away until they converge, this way overcoming the aperture problem. The massive parallel structure allows for and demands a real hardware implementation of the system.

1 Introduction

There are several applications which require a dense displacement field of pixels between images of a video sequence or between different camera images. The efficient and accurate computation of this so-called optical flow especially in presence of occlusion is still an open research topic. Standard dense matchers usually assume some region (a window) around each pixel to be invariant and compare the window of the pixel in image A to all candidate windows in image B to compute a similarity. This creates the problem of finding an appropriate window size, such that the region is significant (aperture problem) but still invariant (regarding occlusion and perspective), which may be avoided in parts if adaptable windows are used [1]. In our approach, there is no need to choose a window size since the region used for matching is automatically extended, when the local information does not suffice for a stable result.

Nearly all proposed optical flow computation methods make use of some kind of smoothness constraint on the flow field as has early been proposed by Horn and Schunck [2], where the actual implementation varies, e.g. in Markov Random Field (MRF) approaches [3], nonlinear diffusion [4], global minimization with discontinuity punishment [5] and so on. However, in addition to adding stability to the matching, smoothing also blurs discontinuities in the flow field. Especially at the projections of 3d object boundaries one would like to have a flow field, which is as sharp as possible, such that smoothing has to be avoided in these regions. Most previous Bayesian methods, which have the advantage of making all assumptions explicit (e.g. [6, 7]) did not take occlusion into account yet. When occluded pixels have been considered (as in [8, 9, 4]), they are usually modelled as some additional disturbance of a prior or yield another penalty term in an error function. In our novel approach they are stated in terms of existing probability distributions in the model. Opposed to that in [10], an algorithm for rectified stereo images only, occlusions and discontinuities are modelled by additional stochastic processes with parameters, which have to be estimated simultaneously to the flow in a global optimization scheme. If explicit knowledge about the number of moving objects is present, the computation of boundary curves around regions with uniform displacement may be possible [11], which can then also be used to avoid smoothing at discontinuities. Other techniques directly using high image gradient values for diffusion blocking discard the significant structure for optic flow estimation contained in these regions. In [9] the prior allows discontinuities only at intensity edges, otherwise penalties apply.

The basic concepts of the algorithm used have already been proposed in [12], but occlusion (and the information it carries) has not been taken into account yet. Furthermore, the concept has been extended regarding a scale pyramid initialization, which makes detailed knowledge on pre-positioning obsolete. Initially, the algorithm computes a probability distribution for each pixel in frame 0 among all (predefined) possible match candidates in frame 1 (a test patch) by calculating the probability that both pixels correspond only based on their color. Using only color information of single pixels is usually not enough, such that an additional constraint has to be imposed on the neighborhood. Thus, a local

probability measure that demands similar displacement for neighboring pixels is iteratively applied to propagate information throughout the image, such that local constraints are transformed into global information over time. The entropy of the probability distribution among match candidates in the test patches is reduced step-by-step and finally converges to a single (sub-)pixel position, which represents the expected correspondence.

In [4] concepts are comparable to our approach but differ in the implementation of the pixel invariance properties. Where they make use of a MRF to enforce a smooth disparity space, we follow the idea that the distribution of correct pixel matches can locally be described by a particular probability distribution, whereas wrong match candidates are uniformly distributed. Based on our model, we derive that the information propagation (as the equivalent of smoothing) is blocked proportionally to the probability that a pixel is occluded either in the first or in the second image. This helps to improve the matching quality between subsequent frames and sharpens the flow field at depth discontinuities, since moving objects usually introduce occluded pixels into the images.

The occlusion probability is derived straight-forwardly from the Bayesian correspondence theory framework, as opposed to more ad hoc occlusion detection algorithms like "goodness of match discontinuities" or "bimodalities in disparity", which are compared in [13]. Nonetheless, detecting occluded areas is an intricate problem, since they may not exactly coincide with the pixel grid, such that they are smoothed into neighboring pixels. The concept of an occlusion probability accounts perfectly for this. This is an improvement over previous methods, which used a binary distinction between occluded and visible pixels during their minimization scheme [9, 8].

The spread of correspondence information implicitly starts at heavily structured regions, which may be viewed like seed crystals and progresses to homogeneous regions, where the probability distributions converge over time. This may be regarded as automatic feature selection and matching at descriptive points and subsequent guided interpolation across less-structured regions. However, no minimization of a global probability model is carried out, since at each iteration step probabilities are updated with local information only.

The algorithm works on uncalibrated image sequences and is much more scale and rotation tolerant than standard window correlation approaches. It can also be used for (uncalibrated) stereo scenarios and is easily adaptable to exploit the knowledge of rectified epipolar geometry, too. Due to its simple structure a huge number of basic operations is necessary to compute the correspondences, which is well-suited for a parallel implementation using specialized hardware like an FPGA (Field Programmable Gate Array) chip or a graphics card.

2 The Bayesian Model

In the model we develop, we are not interested in the exact camera geometry. We simply assume that we are given two images A and B whose pixels are correlated in as far as they represent the same scene, albeit from a different point of view

(stereo matching) or at a different time (optical flow). The only constraints we can invoke then are pixel similarity and an ordering constraint.

We assume that correct matches satisfy a particular statistical distribution whereas incorrect matches are equivalent to noise and are uniformly distributed. We are looking for an iterative procedure that amplifies those pixels that satisfy the appropriate distribution and subdues the others. We can only give a short overview of the algorithm's derivation here. For a detailed account see [14].

First we will derive the local match probabilities, which refer only to the first-order neighborhood of the pixels under inspection. After explaining the concept of correspondence probability, we will finally bind the local probability measures into an iterative algorithm, which increases the region upon which probability statements are based.

2.1 Local Probabilities

The correspondence problem is modelled through random variable pairs (X_A, X_B), where X_A can take on all pixel positions (represented by $\mathcal{I}$) in image A and X_B can take on those in B. However, given some images, these random variables are not independent of one another, we explicitly accept only outcomes of X_A and X_B where the pixel positions $\mathbf{x}_A$ and $\mathbf{x}_B$ both correspond to the same element in 3d space, i.e. the event $(X_A = \mathbf{x}_A, X_B = \mathbf{x}_B)$ means that position $\mathbf{x}_A$ in image A corresponds to $\mathbf{x}_B$ in B. This is our *correspondence pair assumption*, which is implicitly stated in every subsequent probability further down.

To find out which pixels refer to which others, we basically need a measure for pixel similarity. This measure has to express the likelihood that two pixels were created by the same element in a scene, without taking into account any neighboring pixels. Such a measure therefore will be based on a pixel's color, but may also include any other local property like the local scale or local phase. We will denote this measure by $s(a, b)$, where a denotes a pixel color (in image A) and b another color (in image B). A good similarity function is the maximum likelihood estimator as used by Belhumeur in [10].

$$P(A|_{\mathbf{x}_A} = a, B|_{\mathbf{x}_B} = b \;\mid\; X_A = \mathbf{x}_A, X_B = \mathbf{x}_B) \simeq s(a, b) \tag{1}$$

The $\simeq$ here means equality up to a scalar factor, since a pdf must sum to unity. Using $s(A|_{\mathbf{x}_A}, B|_{\mathbf{x}_B})$, we can evaluate for each pixel in image A its similarity to the pixels within an area of image B where we expect the correct match to lie. We will also call this a *test patch* $\mathcal{T}$. Next we scale the computed similarities in $\mathcal{T}$ in a way that they sum to unity, such that we can interpret them as probabilites. That is, each pixel in image A has associated with it a probability distribution giving its matching likelihood to a set of pixels in image B. Our goal is to minimize the entropy of these probability distributions, i.e. to minimize the match uncertainty. In order to do this, the pixel similarity measure alone is not enough. We also have to take into account a structural constraint. We do this by assuming that the local distribution of pixel matches takes on a particular form. This becomes the prior distribution in our derivation, denoted

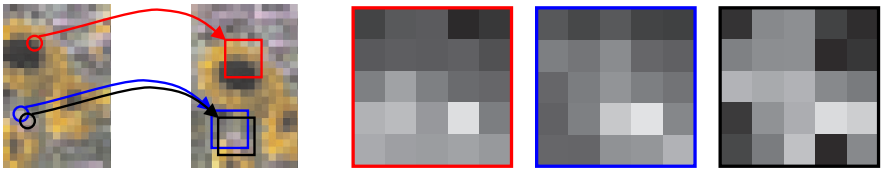

Fig. 1. Left two images: Positions of test patches, right three images: Probability distributions in patches based on pixel similarity

by $h(\mathbf{x}_A, \mathbf{x}_B, \mathbf{y}_A, \mathbf{y}_B)$. That is, given an assumed pixel match $(\mathbf{x}_A, \mathbf{x}_B)$ and a particular neighbor $\mathbf{y}_A$ of $\mathbf{x}_A$, $h(\mathbf{x}_A, \mathbf{x}_B, \mathbf{y}_A, \mathbf{y}_B)$ gives the a priori probability distribution for $\mathbf{y}_B$ being a correct match of $\mathbf{y}_A$. Note that h does not depend on the images but reflects our assumption that neighboring pixels have a similar displacement. Therefore h yields the highest probabilities if $\mathbf{y}_B$ is chosen such that $\mathbf{x}_A - \mathbf{x}_B = \mathbf{y}_A - \mathbf{y}_B$ and lower values the more the lhs and the rhs of this equation differ. Therefore, a simple choice for h may be a zero-mean Gauss of the difference of lhs and rhs of the above.

$$P(Y_B = \mathbf{y}_B \mid X_A = \mathbf{x}_A, X_B = \mathbf{x}_B, Y_A = \mathbf{y}_A) \simeq h(\mathbf{x}_A, \mathbf{x}_B, \mathbf{y}_A, \mathbf{y}_B) \tag{2}$$

This is similar to a Gibbs potential as for example used in MRF approaches [3] to describe the preferences of a disparity surface. However, h does not account for all cliques on the disparity surface as in [4] but only for one neighbor.

We are asking how likely it is that $(\mathbf{x}_A, \mathbf{x}_B)$ is a correct match. This depends on the pixel similarity $(s(A|_{\mathbf{x}_A}, B|_{\mathbf{x}_B}))$ and the likelihood that the eight directly neighboring pixels of $\mathbf{x}_A$, denoted by $\{\mathbf{y}_A^i\}$, have high pixel similarities with those pixels in image B $\{\mathbf{y}_B^i\}$ where $h(\mathbf{x}_A, \mathbf{x}_B, \mathbf{y}_A, \mathbf{y}_B)$ is maximal.

Formally, let (X_A, X_B) and (Y_A, Y_B) be the random variables of two neighboring pixel correspondences, i.e. for some $X_A = \mathbf{x}_A$ only outcomes $\mathbf{y}_A$ of Y_A are inspected, where $\mathbf{x}_A$ and $\mathbf{y}_A$ are neighbors and $\mathbf{x}_A$ and $\mathbf{x}_B$ (and also $\mathbf{y}_A$ and $\mathbf{y}_B$) are corresponding pixels. Using Bayes' law together with the s and h functions we get:

$$P(X_B = \mathbf{x}_B, Y_B = \mathbf{y}_B, Y_A = \mathbf{y}_A \mid A, B, X_A = \mathbf{x}_A) \simeq \tag{3}$$
$$\frac{s(A|_{\mathbf{x}_A}, B|_{\mathbf{x}_B}) s(A|_{\mathbf{y}_A}, B|_{\mathbf{y}_B}) h(\mathbf{x}_B, \mathbf{y}_B, \mathbf{x}_A, \mathbf{y}_A)}{P(X_A = \mathbf{x}_A \mid A, B)}$$

Given some position $\mathbf{x}_A$ in A this equation expresses the probability, that $\mathbf{x}_B$ is the correct match, while the neighbor $\mathbf{y}_A$ of $\mathbf{x}_A$ corresponds to $\mathbf{y}_B$. To make the pdf independent of some particular match candidate $\mathbf{y}_B$, the best matching $\mathbf{y}_B$ (regarding equation (3)) is assumed to be the match of $\mathbf{y}_A$ and its match information is used to evaluate the neighborhood and similarity constraint for $\mathbf{y}_A$. This is in contrast to MRF field methods, which would marginalize at this point over Y_B. We explicitly select only that match candidate for each neighbor, where the image data best satisfies the assumed prior distribution. It was found that this improves convergence. The slightly different pdf is denoted by $\hat{P}$.

Additionally, the match probability for $\mathbf{x}_B$ should also be stated independently of a particular neighbor $\mathbf{y}_A$ of $\mathbf{x}_A$. Therefore we maginalize over all neighbors $\mathbf{y}_A$ and end up with the *pixel-match pdf*:

$$\hat{P}(X_B = \mathbf{x}_B \mid A, B, X_A = \mathbf{x}_A) \simeq$$

$$\frac{s(A|_{\mathbf{x}_A}, B|_{\mathbf{x}_B})}{P(X_A = \mathbf{x}_A \mid A, B)} \cdot \sum_{\mathbf{y}_A : (\mathbf{y}_A - \mathbf{x}_A) \in \mathcal{N}} \max_{\mathbf{y}_B} (s(A|_{\mathbf{y}_A}, B|_{\mathbf{y}_B}) h(\mathbf{x}_A, \mathbf{x}_B, \mathbf{y}_A, \mathbf{y}_B)) \quad (4)$$

With this equation, we can now compute the probability of each candidate match $\mathbf{x}_B$ in B to be the correct match, given some position $\mathbf{x}_A$ in image A. It is only based on the color and the fitting of the direct neighbors.

2.2 Occlusion Detection

Now we want to inspect the probability that a pixel really has a correspondence in the other image at all. One finds that this information is carried in the probability distributions $P(X_A \mid A, B)$ and $P(X_B \mid A, B)$. We will refer to them as the *correspondence probabilities*. Remember that $P(X_A = \mathbf{x}_A \mid A, B)$ is the probability that random variable X_A has the outcome $\mathbf{x}_A$ under the correspondence pair assumption. If there is no correspondence, X_A will never have the outcome $\mathbf{x}_A$. On the other hand, if every pixel in the image has exactly one correspondence, X_A will be uniformly distributed.

To unterstand the principle used for occlusion detection, suppose two images are matched. If everything works as expected, the test patches eventually have a strong peak at the correct position. Now suppose that there are some occluded pixels, e.g. a pixel $\mathbf{x}_O$ in A, which has no correspondence in image B. Nevertheless, its test patch may have a high value at some position, say $\mathbf{x}_B$ in B, which represents the most likely match for $\mathbf{x}_O$. However, if $\mathbf{x}_B$ has a true match $\mathbf{x}_A$ in A, $\mathbf{x}_B$'s test patch will most likely be maximal at this position and vice versa. That is, $\mathbf{x}_O$'s position (in the test patch of $\mathbf{x}_B$) has a very low probability. One might say that $\mathbf{x}_O$ chose $\mathbf{x}_B$ as a correspondence partner but does not get support from the inverse direction. Now, the observation that occluded pixels do not get support from the inverse direction is exploited to detect them. Both, the lhs and the rhs of the following equation represent $P(X_A, X_B \mid A, B)$:

$$P(X_A \mid A, B) P(X_B \mid A, B, X_A) = P(X_B \mid A, B) P(X_A \mid A, B, X_B) \quad (5)$$

This equation must hold for every single match candidate in the test patch. However, instead of inspecting single correspondences, it is more promising to evaluate all possible matches in the other image at once and thus to get more robust information. Equation (5) is solved for $P(X_A \mid A, B)$ and summed across all candidate pixels (i.e. across $\mathbf{x}_A$'s test patch $\mathcal{T}_{\mathbf{x}_A}$):

$$P(X_A = \mathbf{x}_A \mid A, B) = \frac{\sum_{\mathbf{x}_B \in \mathcal{T}_{\mathbf{x}_A}} P(X_A = \mathbf{x}_A \mid A, B, X_B = \mathbf{x}_B) P(X_B \mid A, B)}{\sum_{\mathbf{x}_B \in \mathcal{T}_{\mathbf{x}_A}} P(X_B = \mathbf{x}_B \mid A, B, X_A = \mathbf{x}_A)} \quad (6)$$

Since there is no preference on matching from A to B or from B to A, the same can be applied to the B to A direction analogously. It can be seen that the probability $P(X_A \mid A, B)$ depends on the support of the inverse direction. Unfortunately, it does also depend on the correspondence probability of that direction. Since both probabilities depend on one another, they cannot be calculated explicitly before the start of the algorithm. Instead, each correspondence probability has to be initialized with some value and is updated iteratively utilizing the above equations, which is referred to as the *collection of support*.

We will now give an interpretation of the correspondence probability values for a binary occlusion classification (which may be desired by a high level application). Since $P(X_A = \mathbf{x}_A \mid A, B)$ is a probability (of the event $\mathbf{x}_A$), summing it over all possible $|\mathcal{I}|$ (disjoint) events must yield unity. As pointed out before, if every pixel has a match, $P(X_A \mid A, B)$ must be a uniform distribution with the value $1/|\mathcal{I}|$. If a single pixel $\mathbf{x}_O$ in A is occluded, the true distribution for X_A yields slightly higher values for all pixels having correspondences and zero for $(X_A = \mathbf{x}_O)$. The more occluded pixels exist, the more the correspondence probability for pixels with true matches increases. Accounting for noise and spurious support, a pixel $\mathbf{x}_O$ should therefore only be classified as occluded, if $P(X_A = \mathbf{x}_O \mid A, B) \ll 1/|\mathcal{I}|$.

Intrinsically the correspondence probability is some kind of measure for the support from the inverse direction. Apart from detecting occluded pixels it also yields a low value in case of low similarity (and great uncertainty) for true matches. Hence, it can also be a hint for the confidence in a pixel's match value, indicating how well the pixel and the neighborhood are found in the other image.

2.3 Propagation of Local Constraints

The structural constraint and the pixel-match pdf refer only to direct neighbors of a pixel. Usually more global information is needed for stable matching results, since the aperture problem is very relevant for these small neighborhoods. This section binds the derived equations into an iterative algorithm, where local information propagates throughout the images step by step.

The previously derived pdf (4) is used as the base equation, where the match probability resulting from the last iteration is used as the pixel similarity for the next round. To abstract from the s-function, f^t is defined to contain the similarities from the t^{th} iteration, where the first f is the similarity s. The functions c_A and c_B represent the correspondence probabilities for the pixels at each iteration (initialized with their expectation value).

$$f^0(\mathbf{x}_A, \mathbf{x}_B) := s(A|_{\mathbf{x}_A}, B|_{\mathbf{x}_B}) \qquad c_A^0(\mathbf{x}_A) := 1/|\mathcal{I}| \qquad c_B^0(\mathbf{x}_B) := 1/|\mathcal{I}|$$

Let $\mathcal{F}^t$ contain the information available at iteration t, i.e. f^t, c_A^t and c_B^t. The resulting pdf $\hat{P}(X_B = \mathbf{x}_B \mid \mathcal{F}^t, X_A = \mathbf{x}_A)$ can then be computed up to scale as:

$$\frac{f^t(\mathbf{x}_A, \mathbf{x}_B)}{c_A^t(\mathbf{x}_A)} \sum_{\mathbf{y}_A : (\mathbf{y}_A - \mathbf{x}_A) \in \mathcal{N}} \max_{\mathbf{y}_B \in \mathcal{T}_{\mathbf{y}_A}} \left(f^t(\mathbf{y}_A, \mathbf{y}_B) h(\mathbf{x}_A, \mathbf{x}_B, \mathbf{y}_A, \mathbf{y}_B) \right) \qquad (7)$$

The correspondence probabilities are computed as described in equation (6):

$$c_A^{t+1}(\mathbf{x}_A) = \hat{P}(X_A = \mathbf{x}_A \mid \mathcal{F}^t) \simeq \frac{\sum_{\mathbf{x}_B \in \mathcal{T}_{\mathbf{x}_A}} \hat{P}(X_A = \mathbf{x}_A \mid \mathcal{F}^t, X_B = \mathbf{x}_B) c_B^t(\mathbf{x}_B)}{\sum_{\mathbf{x}_B \in \mathcal{T}_{\mathbf{x}_A}} \hat{P}(X_B = \mathbf{x}_B \mid \mathcal{F}^t, X_A = \mathbf{x}_A)} \tag{8}$$

To get an idea of the final step, imagine first that we directly reuse the probabilities of the last iteration:

$$f^{t+1}(\mathbf{y}_A, \mathbf{y}_B) = \hat{P}(Y_A, Y_B \mid \mathcal{F}^t) \simeq \hat{P}(Y_B = \mathbf{y}_B \mid \mathcal{F}^t, Y_A = \mathbf{y}_A) c_A^t(\mathbf{y}_A) \tag{9}$$

Using this in equation (7), the term $c_A^t(\mathbf{y}_A)$ is independent of the maximization and may be moved in front of it. It is plain to see that all eight neighbours of $\mathbf{x}_A$ are weighted by their correspondence probability c^t, i.e. occluded pixels have a smaller weight than pixels with support. Consequently, the probability for $\mathbf{x}_A$ mainly depends on the probabilities of its not-occluded neighbors.

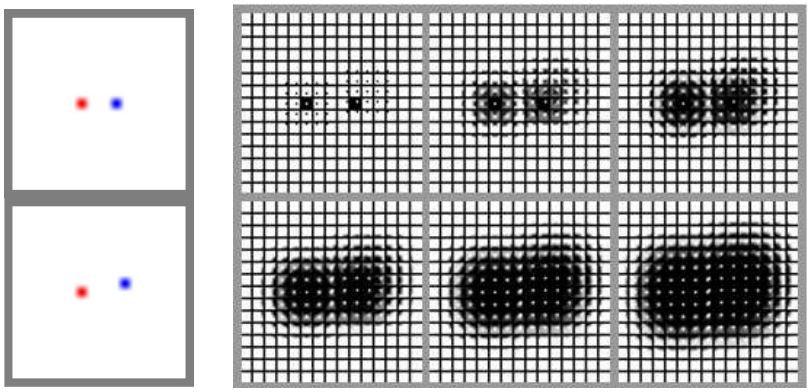

Fig. 2. Upper left: image A, lower left: image B, where blue (right) pixel is moved one pixel right and one up. Right images: Initial test patches (similarity) and first five iterations. At first, test patches in homogeneous regions hold uniform distributions, while those of the red and the blue pixel have already converged to a single position. The neighboring patches have only excluded some candidates. When iterating, more and more candidates becomes improbable, and information propagates throughout the image. However, for the white pixels in between the colored ones, two maxima survive, since it is not clear how they have behaved. The expectation value yields a subpixel position in between the two possible movements.

For stability reasons the simple step of equation (9) is replaced by a bidirectional merging step, because instead of matching from A to B we may also match from B to A as well. Since $P(X_A \mid A, B)P(X_B \mid A, B, X_A)$ and $P(X_B \mid A, B)P(X_A \mid A, B, X_B)$ both represent the same joint probability, they should be equal then. Therefore, we assign their (geometric) average to both of them for the next iteration. Again, this applies only up to scale, since we have to normalize the test patches afterwards. For each patch we choose the factor which makes

it sum to unity. Note that the feature of diffusion control by the correspondence probability is not affected by the bidirectional merging.

$$f^{t+1}(\mathbf{x}_A, \mathbf{x}_B) = \hat{P}(X_A, X_B \mid \mathcal{F}^t) \simeq \tag{10}$$

$$\sqrt{\hat{P}(X_A = \mathbf{x}_A \mid \mathcal{F}^t, X_B = \mathbf{x}_B) c_B^t(\mathbf{x}_B) \hat{P}(X_B = \mathbf{x}_B \mid \mathcal{F}^t, X_A = \mathbf{x}_A) c_A^t(\mathbf{x}_A)}$$

To support large disparities and to detect occlusions at larger scales, the whole process is done using a Gauss pyramid. Starting at the highest layer (usually with image size in the order of 32) images are matched with a test patch size of 5x5 pixels. The pyramid layer and the test patch size encode the maximum displacement expected between the images. For optic flow sequences it is usually sufficient to go up one or two layers, since displacements are in the range of a few pixels. Having calculated the test patches and correspondence probabilities of one layer, test patches of the lower scale are positioned at the interpolated expectation values of the higher ones and correspondence probability is also interpolated from that scale. The matching and occlusion detection is then started for the lower pyramid layer. Depending on the images, after about 10-20 iterations all test patches have converged and the procedure can be repeated with the next layer until finally the original images are matched.

3 Experiments

Though the algorithm allows for a hardware implementation, it has been realized in software for qualitative analysis. Since all operations that may be executed in parallel have to be serialized using a standard CPU, the matching is very slow and takes several minutes. However, an FPGA implementation of a simplified version of the basic algorithm has shown that if enough hardware resources are available, the parallel structure can be exploited to increase speed by several orders of magnitude. To check how the model works in practice, some artificial images with exact ground truth data are evaluated as a first proof of concept in figure (3). To generate image A, a small image is inserted into a large one at a position near the center, simulating a rectangular object, which hides parts of the background. In image B, the foreground object is moved by two pixels to the right and one up, so that it hides a slightly different area of the background.

For evaluation purposes only the A to B direction displacement images are shown, but all observations are also valid for the B to A direction. Setting all correspondence probabilities to constant values (as assumed in [12]) instead of computing them (see figure (4)), most pixels are matched correctly (test patch size 5x3, 20 iterations), but problems occur at the borders of the foreground object. Background pixels left to it and below it are matched badly, although they have well-defined correspondences in the other image. As pointed out, occluded pixels (right of and above the foreground object) have no correspondences, so their match values are neglected here. The displacement field has been smoothed at the left and lower object boundary: Pixels of the background have been strongly influenced by foreground object pixels and are matched as if they were moving

Fig. 3. Artificial flow sequence images (noisy images) generated by moving a 8x16 pixel block (foreground object) by 2 pixels right and one up in front of a 32x32 pixel image (containing uniformly distributed noise,pixelwise independent). Ground truth optical flow is displayed right of each image, i.e. black pixels have the same position in both images, grey ones are displaced. Pure white pixels are occluded in the other image.

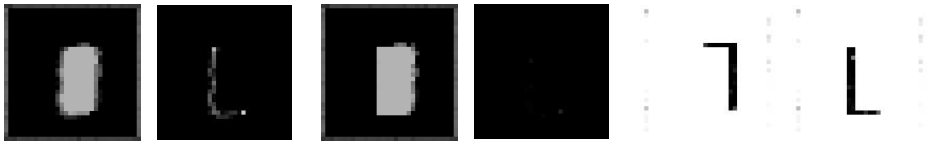

Fig. 4. Left two images: A to B results for disabled occlusion detection: Disparity (left), error (right). Pixels above and to the right of the foreground object are occluded in that direction and therefore disregarded in the error analysis. Middle two images: A to B results using correspondence probability: Disparity (left), error (right). Right two images: Correspondence probability. A to B (left) and B to A (right), black pixels mean low values (probably occluded)

with the object. This occurs due to the structural constraint, that demands similar displacements for neighboring pixels. Since there is a very high degree of information and pixel color differs noticably for wrong match candidates, pixels should be matchable quite uniquely. However, without occlusion information, neighboring pixels belonging to a different object are trusted exactly as much as really neighboring pixels are. The matching error is high in these border regions, as can be seen in the error image in the left part of figure (4).

Now the correspondence probability is concurrently computed using the same parameters as before on the same images. The images show the probabilities that some pixel in A has a correspondence in B (left image) and vice versa (right image). All occluded pixels have been detected, i.e. their correspondence probabilities are low. The implications of correct occlusion detection on the matching result can be seen right in figure (4). There are sharp displacement field discontinuities at the object's lower and left borders. The other borders are not relevant for this matching direction since the pixels right and on top of the foreground object in A are occluded in B.

At first sight it is somewhat surprising that the matching results have improved for the A to B direction in regions where there are actually no occluded pixels in that direction, whose influence could have been reduced. This is achieved by bidirectional merging. Occlusions from the other direction discouple the flow field and optimize the matching accuracy of this direction, too.

Adding some noise to image A and image B independently does not disturb the matching results and occlusion detection as can be seen in table (1). Viewing

Noise Level	disabled occl. detection e_M / $\frac{1}{1000}$	enabled occl. detection e_M / $\frac{1}{1000}$
0 %	34.1	1.6
3 %	34.3	1.7
5 %	34.3	1.7
10 %	36.3	1.6

Table 1. Mean matching error in presence of Gaussian noise (in percent of the dynamic range of the pixel values) added independently to both images

	Simple	Extended	Value Range from [15]
$\bar{e}_m$	0.27	0.21	0.16 .. 0.45
$\bar{e}_{ma}$	0.17	0.13	0.11 .. 0.48

Table 2. Results for the street scene: basic algorithm without occlusion detection(left column), enabled occlusion detection and flow constraint(middle), values of comparing paper(right). $\bar{e}_m$ is the mean absolute error and $\bar{e}_{ma}$ is the mean absolute error in direction orthogonal to the local gradient

the mean matching error in thousandths can give approximately the number of pixels being matched really wrong. Note that the images used have two artificial properties, which are not always valid for natural images: First, there is very strong structure present, which simplifies the matching and the occlusion detection. Usually natural images are smoother and contain regions with low contrast. Secondly, there are no subpixel correspondences, i.e. no probabilities are distributed across some neighboring pixels and there is a one-to-one correspondence. However, in that scenario, occlusion detection works and improves matching results at image borders, even in the presence of noise.

An interesting approach for the evaluation of optic flow algorithms is used in [15], where a ray-tracer is used to render semi-artificial scenes that look more realistic than line patterns or random dot images, but for which full ground truth is provided. We have made experiments on the street sequence from [15], which are quite promising. To exploit the fact that in dense image sequences only small changes apply, we incorporated four frames into the similarity function instead of two. Additionally to using frame 1 and 2 we extrapolate a pixel position in frame 0 and 3 and use their similarities (with smaller weights) as well. This is a preliminary solution, a better way to exploit dense frames may be the use of a Kalman or Particle Filter over time and should be subject to future research. The mean errors are calculated taking into account all images of the sequence and can be seen in table (2) as well as the error range from [15]. The algorithm performs quite well compared to other optic flow algorithms used for this sequence regarding the mean error e_m, again, occlusion information improves the matching result. McCane et al. state that the value e_{ma}, which is quite low here, can give a hint about how well the algorithm can cope with the aperture problem. This value refers only to the first order neighborhood of some pixel and does not depend on the absolute value of the gradient. However, regarding that measure the algorithm can handle the aperture problem quite well, since the error vectors do not always point into the locally most uncertain direction.

The next example shows the results of the aerial Pentagon pair provided by CMU/VASC. Note that the exploitation of the epipolar geometry is only that the test patch height can be set to one. Apart from that the images are handled as if they were two frames of a video sequence.

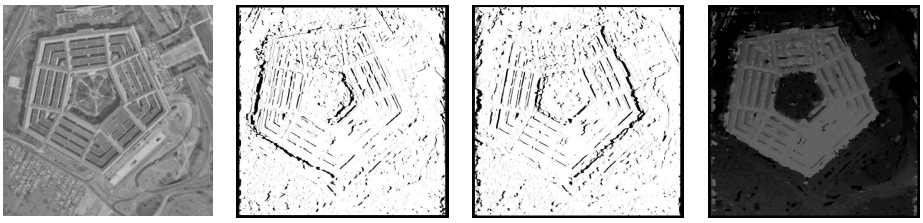

Fig. 5. Aerial image from the CMU/VASC Pentagon pair (left), correspondence probability (middle images) and absolute value of optic flow (right)

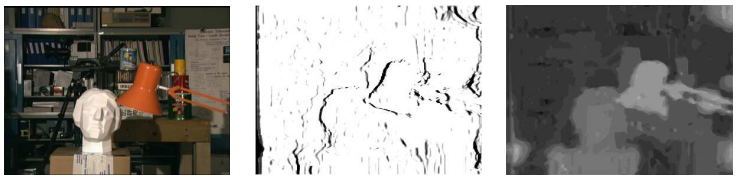

Fig. 6. Image of Tsukuba stereo pair, correspondence probability, computed disparity

As pointed out before, white pixels (in the middle images of figure (5)) mean great correspondence probability while black pixels mean low support and thus indicate occluded pixels. Note that the main occlusion lines are detected, but that these lines are not absolutely accurate and sharp. Hence, the matching results are also not sharpened as much as it may have been expected. For completeness reasons the matching results are displayed in the disparity image of figure (5), although this example is intended to demonstrate the occlusion detection in real images. Since there is no real ground truth available for this image pair, it is not exactly known, where the occlusions are and how they look like. If a bird flies over the building or some tree is moving in the wind, occlusion changes. However, the detected occlusions refering to the building look pretty realistic.

In figure (6) the well known Tsukuba pair is displayed. The correspondence probability images show that the algorithm can extract occluded pixels even in real images with (partly) weak structure beneath object borders. Though these probability images are quite noisy, the main occlusions can be seen well. It is also interesting to see that the correspondence probability image is black at the left image border, which is no error due to border problems. Quite the reverse, these pixels are also occluded in the other image, since they have no correspondence there.The disparity image of figure (6) shows a good matching result compared to the ground truth image. The scale approach has positioned the test patches well and the depth discontinuities are represented in the disparity image.

4 Conclusion

A dense matching algorithm has been proposed, which is based on explicit assumptions about local probability distributions in the images. The algorithm extracts occlusion probability for each pixel, which is used in the model to steer the diffusion process. We have shown that this increases the matching quality significantly at depth discontinuities. The algorithm works for uncalibrated cameras and supports a wide range of displacements, since it matches and detects occlusion over a scale pyramid. No explicit window size parameter is needed since the aperture problem is handled automatically by information propagation from structured to homogeneous regions over time, which can be interpreted as a recurrent neural network converging to a stable state. Due to its fundamentally parallel structure, a fast hardware implementation is necessary, which has not been done so far. Another focus of future research must be the exploitation of multiple frames to stabilize the matching and occlusion extraction process.

References

1. Kanade, T., Okutomi, M.: A stereo matching algorithm with an adaptive window: Theory and experiment. IEEE Transactions on PAMI **16** (1994) 920–932
2. Horn, B., Schunck, B.: Determining optical flow. AI **17** (1981) 185–204
3. Marroquin, J., Velasco, F., Rivera, M., Nakamura, M.: Gauss-markov measure field models for low-level vision. IEEE Transactions on PAMI **23** (2001) 337–348
4. Scharstein, D., Szeliski, R.: Stereo matching with nonlinear diffusion. International Journal of Computer Vision **28** (1998) 155–174
5. Zitnick, C., Kanade, T.: A cooperative algorithm for stereo matching and occlusion detection. IEEE Transactions on PAMI **22** (2000) 675–684
6. Vasconcelos, N., Lippman, A.: Empirical bayesian motion segmentation. IEEE Transactions on PAMI **23** (2001) 217–221
7. Torr, P., Szeliski, R., Anandan, P.: An integrated bayesian approach to layer extraction from image sequences. IEEE Transactions on PAMI **23** (2001) 297–303
8. Black, M., Rangarajan, A.: On the unification of line processes, outlier rejection, and robust statistics with applications in early vision. IJCV **19(1)** (1996) 57–91
9. Konrad, J., Dubois, E.: Bayesian estimation of motion vector fields. IEEE Transactions on PAMI **14** (1992) 910–927
10. Belhumeur, P.N.: A Bayesian approach to binocular stereopsis. International Journal of Computer Vision **19** (1996) 237–262
11. Memin, E., Perez, P.: Dense estimation and object-based segmentation of the optical flow with robust techniques. IEEE Transactions on IP **7** (1998) 703–719
12. Perwass, C., Sommer, G.: An iterative bayesian technique for dense image point matching. In: Proceedings of Dynamic Perception. (2002) 283–288
13. Egnal, G., Wildes, R.: Detecting binocular half-occlusions: Empirical comparisons of four approaches. In: IEEE Conference CVPR. (2000) 466–473
14. Koeser, K.: Dense image point matching with explicit occlusion detection for stereo disparity and optic flow. Master's thesis, CAU Kiel (2003) (available at http://www.ks.informatik.uni-kiel.de)
15. Galvin, B., McCane, B., Novins, K., Mason, D., Mills, S.: Recovering motion fields: An evaluation of eight optical flow algorithms. In: Proc. of BMVC. (1998) 195–204