An iterative Bayesian technique for Dense Image Point Matching

Christian B. U. Perwass[1] and Gerald Sommer[2]

[1] Institut für Informatik, CAU Kiel, 24105 Kiel, Germany
christian@perwass.de, www.perwass.de
[2] Institut für Informatik, CAU Kiel, 24105 Kiel, Germany
gs@ks.informatik.uni-kiel.de

Abstract

We present a conceptually simple algorithm for dense image point matching between two multi-modal (e.g. color) images. The algorithm is based on the assumption that correct image point matches satisfy locally a particular statistical distribution. Through an iterative evaluation of a local probability measure, global constraints are taken into account and the most likely set of image point matches is found. An advantage of this approach is that no information about the camera geometries, as for example the epipoles, has to be known. Therefore, the algorithm can be used for stereo matching and optic flow.

1 Introduction

The basic idea behind all optic flow and stereo matching algorithms is, that if two images are projections of the same 3D-scene taken from slightly different positions or at slightly different times, then certain properties of corresponding pixels are invariant. However, it is not necessarily the case that a pixel in one image can be identified with exactly one pixel in the other, since rigid objects may appear shrunk or grown in different projections. Furthermore, parts of a 3D-scene that can be seen in one projection may be occluded in the other. The transformation between two images related by optic flow or stereo, is therefore more like a homotopy, as Florack et al. [1] point out, than a vector field. Nevertheless, a vector field is what we need in most applications. Therefore, in general an assumption is made about the invariant properties of corresponding pixel, which approximates nearly invariant properties of the underlying homotopy.

The invariant properties which are typically identified are those of pixel color and pixel neighborhood structure. Algorithms differ in how they model these invariances and the method employed in identifying corresponding pixels using the assumed invariant properties.

Some different types of approaches are for example: feature based methods (e.g. [2]), pixel labelling methods (e.g. [3, 4, 5]) and Bayesian methods (e.g. [6, 7, 8, 9]).

Bayesian methods have the advantage of clearly stating the invariance assumptions made about corresponding pixels by defining priors on the parameters of the system. Markov random field (MRF) approaches as described in [10] play an important role in this context [11]. The details of the different Bayesian approaches to dense image point matching are quite varied. However, typically they do not assign a single disparity label to a pixel but a discrete probability distribution function (pdf) over a set of disparities. Although this might, at first, seem to violate the often used uniqueness assumption as stated by Marr and Poggio [12], one can always define the final disparity to be the expectation value of the pdf. The advantage of defining a discrete pdf is that, in effect, we can test a number of hypotheses concurrently and eventually extract the most likely one. Finding the set of disparities which maximizes an appropriately defined probability measure then gives the answer to the correspondence problem. Such a maximization may be done iteratively or through a global maximization scheme.

In this paper we also follow a Bayesian approach which is based on an idea we published previously [13] using different mathematical tools. A detailed discussion of our approach, including a number of experiments, can be found in [14]. Our approach is similar to [15] but differs in the implementation of the pixel invariance properties. Where they use a MRF approach to enforce a smooth disparity space, we follow the idea that the distribution of correct pixel matches can locally be described by a particular pdf, whereas wrong match candidates are uniformly distributed. Through an iterative evaluation of a local probability measure, local matching constraints are propagated through the image, such that global constraints are taken into account. Although, occlusion is not modelled explicitly, half-occluded pixels are either given two different disparities simultaneously, or they are matched onto the nearest matchable pixel. That is, the algorithm does not break down in the presence of occlusion.

2 Theory

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 pixel 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 image point 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 pixel 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 of all we 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 $s(\mathbf{x}_A, \mathbf{x}_B)$, where $\mathbf{x}_A$ denotes a pixel position in image A and $\mathbf{x}_B$ a pixel position in image B.

Using $s(\mathbf{x}_A, \mathbf{x}_B)$, 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. 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 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 $\mathbf{y}_A$.

It can be shown that the probability of $(\mathbf{x}_A, \mathbf{x}_B)$ and $(\mathbf{y}_A, \mathbf{y}_B)$ being two neighboring pixel matches is then given by

$$\begin{aligned} &P(\mathbf{X}_B = \mathbf{x}_B, \mathbf{Y}_B = \mathbf{y}_B | A, B, \mathbf{X}_A = \mathbf{x}_A, \mathbf{Y}_a = \mathbf{y}_A) \\ &\quad = s(\mathbf{x}_A, \mathbf{x}_B) s(\mathbf{y}_A, \mathbf{y}_B) h(\mathbf{x}_A, \mathbf{x}_B, \mathbf{y}_A, \mathbf{y}_B). \end{aligned} \tag{1}$$

The probability measure on which we base our match decision is the following. Assuming $(\mathbf{x}_A, \mathbf{x}_B)$ are a correct match, then for a given neighbor $\mathbf{y}_A$ of $\mathbf{x}_A$ we say that the most likely match $\mathbf{y}_B$ of $\mathbf{y}_A$ is the one where the data best satisfies the prior distribution of neighboring matches. That is we are looking for the estimator $\hat{\mathbf{y}}_B$ given by

$$\hat{\mathbf{y}}_B = \arg\max_{\mathbf{y}_B} \left(\frac{P(\mathbf{X}_B, \mathbf{Y}_B = \mathbf{y}_B | A, B, \mathbf{X}_A, \mathbf{Y}_A)}{\max_{\mathbf{y}} P(\mathbf{X}_B, \mathbf{Y}_B = \mathbf{y} | \mathbf{X}_A, \mathbf{Y}_A)} \right). \tag{2}$$

The effect of this is that if for a particular set $(\mathbf{x}_A, \mathbf{x}_B, \mathbf{y}_A)$ the corresponding $\hat{\mathbf{y}}_B$ maximizes the prior, then

$$P(\mathbf{X}_B = \mathbf{x}_B, \mathbf{Y}_B = \hat{\mathbf{y}}_B | A, B, \mathbf{X}_A = \mathbf{x}_A, \mathbf{Y}_A = \mathbf{y}_A) = s(\mathbf{x}_A, \mathbf{x}_B) s(\mathbf{y}_A, \mathbf{y}_B). \tag{3}$$

That is, the match probability depends solely on the pixel similarities.

What we really need to estimate is the probability of $(\mathbf{x}_A, \mathbf{x}_B)$ being a correct match. However, for each neighbor $\mathbf{y}_A$ of $\mathbf{x}_A$ we obtain a match probability estimate from $P(\mathbf{X}_B, \mathbf{Y}_B = \hat{\mathbf{y}}_B | A, B, \mathbf{X}_A, \mathbf{Y}_A)$. We therefore take the final match probability estimate of a pixel pair $(\mathbf{x}_A, \mathbf{x}_B)$ to be the expectation value of the set of probability estimates for all eight neighbors of $\mathbf{x}_A$.

$$\begin{aligned} &P(\mathbf{X}_B = \mathbf{x}_B \,|\, A, B, \mathbf{X}_A = \mathbf{x}_A) \\ &\quad = \rho\, s(\mathbf{x}_A, \mathbf{x}_B)\, \tfrac{1}{8} \textstyle\sum_{\mathbf{y}_A}\ \max_{\mathbf{y}_B}\ s(\mathbf{y}_A, \mathbf{y}_B)\, \dfrac{h(\mathbf{x}_A, \mathbf{x}_B, \mathbf{y}_A, \mathbf{y}_B)}{\max_{\mathbf{y}} h(\mathbf{x}_A, \mathbf{x}_B, \mathbf{y}_A, \mathbf{y})}, \end{aligned} \tag{4}$$

where ρ is a normalization constant and the sum over $\mathbf{y}_A$ goes over all eight neighbors of $\mathbf{x}_A$.

Evaluating the probability measure from equation (4) only once, will not give us a final match result. In order to minimize the entropy of the match probability distributions, we have to apply this measure iteratively. This distributes local match information throughout the image. It also means that homogeneous areas are matched according to the match constraints contained in their surroundings. In [14] we have shown that such an iteration converges. Note that this iterative procedure can be regarded as a recurrent neural network, whose equilibrium state gives the match result.

Half-occluded pixels, i.e. pixels that appear in one image but not in the other, have not been treated explicitly. However, by using a bidirectional matching scheme, the matching process is stabilized in the presence of half-occluded pixels.

3 Experimental Results and Conclusions

Fig. 1. Left image Pentagon example with evaluated disparity map.

In order to run the algorithm we have to set five parameters: the number of iterations to perform, the test patch size, the mean pixel displacement, the standard deviation of the ordering constraint σ_h and the standard deviation of the pixel similarity function σ_s. The mean displacement is basically an approximate pixel match. This is easy to find for optical flow, since we assume that corresponding pixel are almost at the same position. For stereo correspondence this initial match will have to be set by some other means. Finding the best number of iterations could be automated by stopping the algorithm once it has converged. The test patch size has to be set such that the correct match is always included. Here we have to make an assumption about how much we expect the pixels to have moved. The parameters σ_h and σ_s only change details of the final match result. They do not have to be changed for different images.

The Pentagon stereo pair was provided by CMU/VASC. Here we matched an area of 500×500 pixels with a test area size of 21×1. Figure 1 shows the first of the two Pentagon images together with the evaluated disparity map after 20 iterations. It can be seen that the algorithm works quite well for stereo matching on rectified images.

We used the Yosemite sequence created by Lynn Quann at SRI to test the algorithm in an optic flow setting. We matched the lower part of the first two images of the sequence, since no ground truth is available for the cloud region. The image dimensions were 315×177 pixels, the test patch size was 7×7 pixels. The parameters σ_s and σ_h had the same values as in the Pentagon example. We performed 20 iterations which took approximately 150 seconds on an AMD Athlon XP 1800+ (1.53 GHz) running Windows XP. The algorithm runs about twice as fast if we do not perform bidirectional matching, which stabilizes the algorithm in the presence of occlusion. Note that the implementation of the algorithm was experimental and not optimized for speed.

Fig. 2. Initial image of Yosemite sequence.

We evaluated the Euclidean distance between our match results and ground truth. Figure 3 shows the distribution of the pixel match errors over the image. White regions indicate that the pixel match errors are below half a pixel. The next darker level indicates pixel errors of between half and one pixel. The meaning of the other shades of gray are given in the legend of figure 3. Note that since we try to match pixel onto pixel, half a pixel error is as good as we can statistically expect the result to be. Large areas have been matched very well, whereas there are problems in the area of the mountain on the left. Nevertheless, problematic areas are locally confined, which shows the robustness of the algorithm. Recall that we only used two images to evaluate the optic flow. By extending the algorithm to incorporate more images of a flow sequence we hope to improve the matching quality further.

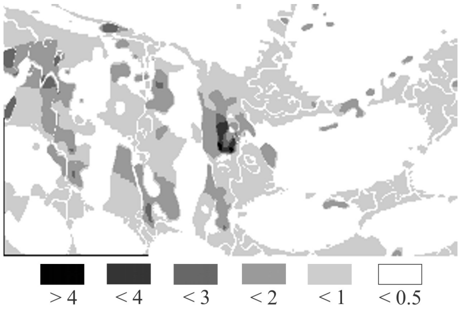

Fig. 3. Distribution of matching errors.

Although the algorithm has a simple mathematical structure, its computational complexity is high. Nevertheless, in principle the match likelihood estimation of all pixels can be done in parallel. In fact, each element of the pixel match probability distributions can be regarded as a single neuron which performs a simple calculation. Evaluating each neuron is all that has to be done per iteration. We have implemented a similar structure on an FPGA which shows good preliminary results.

Of course, there are still a number of problems that have to be addressed by future research. Nevertheless, the results obtained with the algorithm show that despite its simple structure, it is a good dense image point matcher. Note that a program called *Acre* to test the algorithm on arbitrary images, is available from the web page of the first author (`www.perwass.de`).

References

[1] L. Florack, W. Niessen, and M. Nielsen. The intrinsic structure of optic flow incorporating measurment duality. *International Journal of Computer Vision*, 27(3):263–286, 1998.

[2] J. L. Barron, D. J. Fleet, and S. S. Beauchemin. Performance of optical flow techniques. *International Journal of Computer Vision*, 12(1):43–77, 1994.

[3] G. le Besnerais and H. Oriot. Disparity estimation for high resolution stereoscopic reconstruction using the gnc approach. In *Proc. IEEE Int. Conf. on Image Processing*, volume 2, pages 594–597, 1998.

[4] C.L. Zitnick and T. Kanade. A cooperative algorithm for stereo matching and occlusion detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(7):675–684, 2000.

[5] Y. Boykov, O. Veksler, and R. Zabih. Fast approximare energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(11):1222–1239, 2001.

[6] Peter N. Belhumeur. A Bayesian approach to binocular stereopsis. *International Journal of Computer Vision*, 19:237–262, 1996.

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

[8] N. Vasconcelos and A. Lippman. Empirical bayesian motion segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(2):217–221, 2001.

[9] P.H.S. Torr, R. Szeliski, and P. Anandan. An integrated bayesian approach to layer extraction from image sequences. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(3):297–303, 2001.

[10] S. Geman and D. Geman. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(6):721–741, 1984.

[11] J.L. Marroquin, F.A. Velasco, M. Rivera, and M. Nakamura. Gauss-markov measure field models for low-level vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(4):337–348, 2001.

[12] D. Marr and T. Poggio. Cooperative computation of stereo disparity. *Science*, 194:209–236, 1976.

[13] C.B.U. Perwass and G. Sommer. A fuzzy logic algorithm for dense image point matching. In *Proceedings of Vision Interface 2001*, pages 39–47, 2001.

[14] C.B.U. Perwass and G. Sommer. Dense image point matching through propagation of local constraints. Bericht Nr. 0205, Christian-Albrechts-Universität Kiel, 2002.

[15] D. Scharstein and R. Szeliski. Stereo matching with nonlinear diffusion. *International Journal of Computer Vision*, 28(2):155–174, 1998.