Stereo Matching with Non-Linear Diffusion

DANIEL SCHARSTEIN
Department of Mathematics and Computer Science, Middlebury College, Middlebury, VT 05753
schar@middlebury.edu

RICHARD SZELISKI
Microsoft Research, One Microsoft Way, Redmond, WA 98052-6399
szeliski@microsoft.com

Abstract. One of the central problems in stereo matching (and other image registration tasks) is the selection of optimal window sizes for comparing image regions. This paper addresses this problem with some novel algorithms based on iteratively diffusing support at different disparity hypotheses, and locally controlling the amount of diffusion based on the current quality of the disparity estimate. It also develops a novel Bayesian estimation technique which significantly outperforms techniques based on area-based matching (SSD) and regular diffusion. We provide experimental results on both synthetic and real stereo image pairs.

Keywords: Stereo matching, variable-sized support region, non-linear diffusion, Bayesian estimation

1. Introduction

Stereo correspondence is the problem of finding matching points in two or more images of the same scene, usually assuming known camera geometries. Two image points \( p \) and \( p' \) match if they result from the projection of the same point \( P \) in the scene, a property that is often approximated by a similarity constraint requiring, for example, \( p \) and \( p' \) to have similar intensity or color. The desired output of a stereo correspondence algorithm is a disparity map, specifying the relative displacement of matching points between images.

The stereo correspondence problem is inherently underconstrained and further complicated by the fact that the images typically contain noise.

Traditional approaches thus either try to only recover a subset of matches, or make additional assumptions. Feature-based approaches, belonging to the former category, only match points with a certain amount of local information (such as intensity edges), with the disadvantage of yielding only sparse disparity maps. In this paper we will focus on area-based approaches, which yield a dense disparity map by matching small image patches as a whole, relying on the assumption that nearby points usually have similar displacements.

A typical area-based stereo matching algorithm proceeds the following way: For each location in one image, find the displacement that aligns this location with the best matching location in the other image. The quality of a match is measured by comparing windows centered at the two loca-
tions, for example, using the sum of squared intensity differences (SSD).

A more general way of characterizing area-based algorithms is the following:

1. For each disparity under consideration, compute a per-pixel matching cost (e.g., squared intensity difference)
2. Aggregate support spatially (e.g., by summing over a window, or by diffusion)
3. Across all disparities, find the best match based on the aggregated support
4. Compute a sub-pixel disparity estimate (optional)

A central problem is to find the optimal size of the support region (Okutomi & Kanade 1992; Kanade & Okutomi 1994). If the region is too small, a wrong match might be found due to ambiguities and noise. If the region is too big, it can no longer be matched as a whole due to foreshortening and occlusion, with the result of lost detail and blurring (or dislocating) object boundaries in the resulting disparity map.

In this paper, we first review the relevant literature and the basic idea of aggregating support (Sections 2 and 3). We then present some new algorithms that determine the best support region by iteratively diffusing support in a non-linear fashion (Section 4). In Section 5 we develop a Bayesian model of stereo matching using explicit disparity distributions, and derive a novel iterative support aggregation algorithm with significantly improved performance. We present comparative results for our algorithms in Section 6, and close with a discussion of future work.

2. Previous work

In our discussion of related work we will focus on the different processing stages of the area-based algorithm outlined above. A general review of the stereo vision literature is beyond the scope of this paper. For surveys of the field see (Barnard & Fischler 1982; Dhond & Aggarwal 1989).

2.1. Matching cost

At the base of any matching algorithm is a matching cost that measures the (dis-)similarity of two locations. Matching costs can be defined locally (at pixel level), or over a certain area of support. Examples for local costs are absolute intensity differences (Kanade 1994), squared intensity differences (Mathies et al. 1989), binary pixel matches (Marr & Poggio 1976), edges (Baker 1980), filtered images (Marr & Poggio 1979; Jenkin et al. 1981; Jones & Malik 1992), and measures based on gradient direction (Seitz 1989) or gradient vectors (Scharstein 1994). Matching costs that are defined over a certain area of support include correlation (Ryan et al. 1980) and non-parametric measures (Zabih & Woodfill 1994). These can be viewed as a combination of the matching cost and aggregation stages.

2.2. Evidence aggregation

Aggregating support is necessary for stable matching. A support region can either be two-dimensional at a fixed disparity (favoring fronto-parallel surfaces), or three-dimensional in x-y-d space (supporting slanted surfaces). Two-dimensional evidence aggregation has been done using square windows (traditional), Gaussian convolution (Scharstein 1994), multiple windows anchored at different points (Intille & Bobick 1994), and windows with adaptive sizes (Arnold 1983; Okutomi & Kanade 1992; Kanade & Okutomi 1994). Three-dimensional support functions that have been proposed include limited disparity difference (Grimson 1985), limited disparity gradient (Pollard et al. 1985), and Prazdny’s coherence principle (Prazdny 1985), which can be implemented using two diffusion processes (Szeliski & Hinton 1985).

As mentioned above, some techniques, such as correlation and rank statistics which are defined over a fixed support region, can combine the cost and aggregation steps into one. Measures that can be accumulated in a separate step have the following advantages:

- efficiency: the measure can be aggregated with a single convolution (or box-filter) operation (Kanade 1994),
• parallelizability: the aggregation step can be implemented by local iterative diffusion, making the algorithm suited for highly parallel architectures (Szeliski & Hinton 1985);
• adaptability: the measure can be aggregated over locally different support regions using either adjustable size windows (Kanade & Okutomi 1994) or a non-uniform diffusion process (this paper).

2.3. Disparity selection

The easiest way of choosing the best disparity is to select at each pixel the minimum aggregated cost across all disparities under consideration ("winner-take-all"). A problem with this is that uniqueness of matches is only enforced for one image (the reference image), while points in the other image might get matched to multiple points. Cooperative algorithms employing symmetric uniqueness constraints are one attempt to solve this problem (Marr & Poggio 1976). Using dynamic programming techniques (Arnold 1983; Ohta & Kanade 1985; Cox 1994; Intille & Bobick 1994) is another way of selecting unique and consistent disparities. However, these techniques require the strict enforcement of ordering constraints (Yuille & Poggio 1984).

2.4. Sub-pixel disparity computation

Sub-pixel disparity estimates can be computed by fitting a curve to the matching costs at the discrete disparity levels (Lucas & Kanade 1981; Tian & Hulns 1986; Matthies et al. 1989; Kanade & Okutomi 1994). This provides an easy way to increase the resolution of a stereo algorithm with little additional computation. However, to work well, the intensities being matched must vary smoothly.

2.5. Diffusion-based techniques

Non-linear and anisotropic diffusion has been proposed for a variety of early vision tasks, including edge-detection (Perona & Malik 1990; Nordström 1990). Proesmans et al. (1994) detect discontinuities in optical flow by comparing forward and backward flow estimates and then using a diffusion process to smooth the discontinuity maps. (Similar ideas of comparing left-to-right and right-to-left estimates in stereo have also been used by Fua 1993 and others.) Proesmans et al. and Fua also use an anisotropic diffusion process (mediated by intensity gradients) to smooth out the flow/disparity estimates. Shah (1993) has also used non-linear diffusion in the conjunction with a gradient descent algorithm for stereo matching. Shah’s work, however, only models a single disparity at each pixel as opposed to our multiple simultaneous disparity hypotheses.

2.6. Other techniques

Other stereo techniques include hybrid and iterative techniques, such as stochastic search (Szeliski & Hinton 1985; Marroquin et al. 1987; Barnard 1989) and joint matching and surface reconstruction (Hoff & Ahuja 1989; Olsen 1990; Stewart et al. 1996). Hierarchical (coarse-to-fine) matching is another important technique which allows for a larger range of disparities to be matched without excessive search (Quam 1984; Witkin et al. 1987). Yang et al. (1993) use a local winner-take-all strategy within a multiresolution pyramid to find correspondences.

More than two images are used in multiframe stereo to increase stability of the algorithm (Bolles et al. 1987; Matthies et al. 1989; Kang et al. 1995). A special case is multiple baseline stereo, where all images have identical epipolar lines (Okutomi & Kanade 1993). In this case, the similarity measures between the reference image and all other images can be combined by summation into a single measure before the aggregation step.

Finally, occlusion is an important issue. Many approaches ignore the effects of occlusion; others try to minimize them by using a cyclopean disparity representation (Barnard 1989), or try to recover occluded regions after the matching by cross-checking. Several authors have developed methods for dealing with occlusion explicitly, using Bayesian models and dynamic programming (Belhumeur & Mumford 1992; Cox 1994; Geiger et al. 1992; Intille & Bobick 1994).
2.7. Focus of this paper

From the discussion above, it appears that most area-based stereo correspondence algorithms are composed of four tasks: computing a local matching cost; aggregating support spatially; finding the best disparity; and computing a sub-pixel disparity estimate. This framework allows us to compare different approaches that have been taken for each task in isolation, without being distracted by how the other tasks are being solved.

In this paper, we focus mainly on the second task: Aggregating support. We discuss various kinds of local diffusion, including a membrane model and a full distribution model, and contrast it to existing approaches, such as SSD and adaptive windows.

The other three tasks, although important, are not the central issue of this work. Unless noted otherwise, we use squared intensity differences as a matching cost, and, after the aggregation step, simply select the best disparity locally at each pixel. In the cases where we compute sub-pixel disparity estimates, we fit a parabola to the three cost values centered around the best disparity. It is important to keep in mind that the algorithms presented in this paper are independent of these choices and apply also to more sophisticated matching costs and disparity selection strategies.

3. Aggregating support in disparity space and the SSD algorithm

In this section, we introduce the concept of disparity space, review the sum-of-squared-differences (SSD) algorithm, and discuss the need for spatially-adaptive support regions.

3.1. Disparity space

Support for a match is defined over a three-dimensional disparity space $E(x, y, d)$. Formally, we define the initial (not yet aggregated) disparity space $E_0$ as

$$E_0(x, y, d) = \rho(I_L(x + d, y) - I_R(x, y)), \quad (1)$$

where $I_L$, $I_R$, are the intensity functions of the left and right image respectively, and $\rho$ measures the similarity between the two intensities, e.g.,

$$\rho(l - r) = (l - r)^2.$$

This formulation uses $I_R$ as the reference image, and assumes rectified images, i.e., purely horizontal disparities. After aggregating support into a final space $E(x, y, d)$, we can compute a disparity function

$$d(x, y) = \arg\min_{d \in D} E(x, y, d) \quad (2)$$

that represents the matches as offsets to the points in the right image. In practice, we will compute a discrete disparity field

$$d_{i,j} = d(x_i, y_j). \quad (3)$$

Figure 1 illustrates the selection of the best disparity in a vertical disparity column after the aggregation of support at each disparity level. $E$ is a skewed version of the symmetric disparity space $E$ (Marr & Poggio 1976),

$$E(x_R, x_L, y) = \rho(I_R(x_R, y) - I_L(x_L, y)),$$

which reflects that the matching problem is not biased towards either eye. In a symmetric setting, however, it is more difficult to enforce uniqueness for each pixel and to define the final disparity map (see Section 7 for a discussion). Figure 2 illustrates the shape of slices through $E$ and $E$ for a given $y$ and a limited disparity range $D = [d_{\text{min}}, d_{\text{max}}]$.

3.2. SSD

The standard sum-of-squared-differences algorithm (SSD) uses square windows to aggregate the evidence at each disparity. As mentioned before, choosing the right window size involves a trade-off between a noisy disparity map and blurring of depth boundaries. We will illustrate this using two synthetic image pairs. Both pairs have the same disparity pattern (see Figure 3): a central square floating in front of a background with constant disparity. Figure 3 (c) includes the occlusion information: the area displayed in white cannot
be matched due to occlusion, and algorithms will assign arbitrary disparities in this region.

Figure 4 shows the two synthetic image pairs based on this disparity pattern. The first pair, ramp, is similar to the image pair in Fig. 5 in (Kanade & Okutomi 1994) and is based on a linear intensity ramp in the direction of the baseline. Gaussian noise has been added to each image independently. The second image pair, rds, is based on a binary random dot pattern using two gray levels with equal probability. No noise has been added to this image pair.

The two image pairs are quite different. The ramp pair has no local texture variation and constant gradients everywhere, except for the boundaries of the central square. The two images can only be matched by comparing absolute intensities, and any algorithm based on band-pass filtered intensities or gradients will fail (as will the human visual system). The rds pair, on the other hand, has strong local texture variation, but is highly ambiguous since pixels not in correspondance still have a 50% chance of matching.

Figure 5 shows the performance of the simple SSD algorithm on these two image pairs using two different window sizes, \( w = 3 \) and \( w = 7 \). As can be seen, the bigger window size yields a disparity map with less noise, but results in an overall blurring of the features (the “bumpiness” in the recovered disparities is due to sub-pixel disparity estimation). The effect on the two image pairs is quite different: in the ramp pair, the disparities are smoothed across the boundaries, while in the rds pair only the outlines of the square are blurred, i.e., the corners are rounded, while the two disparity levels of foreground and background are clearly recovered.

The latter effect, smoothing of object boundaries, is more common in real images pairs than the smoothing of disparities. The smoothing of disparities we observed in the ramp pair is a di-
rect result of the ramp intensity pattern and the small local variations in intensity.

3.3. The need for adaptive support regions

Let us briefly discuss the reasons for boundary blurring by considering the support for two points $a$ and $b$ inside the central square, but close to its boundary (see Figure 6). Both points receive partial support for the two disparities $d_f$ and $d_b$ of foreground and background respectively, and little support for other disparities. Point $a$, lying next to one of the sides of the square, receives slightly more support from the inside of the square, and is thus correctly found to be at disparity $d_f$. Point $b$, lying in the corner, however, receives more support for $d_b$, since almost $3/4$ of its support region cover the background, and thus is erroneously found to be at disparity $d_b$. The overall effect is that corners get rounded since points close to corners are “co-opted” into the wrong disparity. Straight object boundaries are not affected. Note also that no smoothing of the disparity values takes place.

Since the blurring of outlines is caused by support regions that span object boundaries, a possible solution to the problem is to use non-uniform and adaptive support regions. Kanade and Okutomi (1994) have proposed adaptive windows, square windows that extend by different amounts in each of four directions. The optimal window size is found by a greedy algorithm (gradient descent) based on an estimate of disparity uncertainty in the current window. In this paper we propose a different approach: aggregating support with a non-uniform diffusion process.

4. Aggregating support by diffusion

Instead of using a fixed window, support can also be aggregated with a weighted support function such as a Gaussian. A convolution with a Gaussian can be implemented using local iterative diffusion (Szeliski & Hinton 1985) defined by the equation

$$\frac{\partial E}{\partial t} = \nabla^2 E.$$  \hfill (4)

![Fig. 5. Performance of the SSD algorithm using square windows with sizes $w = 3$ and $w = 7$ on the ramp and rds image pairs.](image1)

![Fig. 6. Support for the two disparities $d_f$ and $d_b$ of foreground and background for two points $a$ and $b$ close to the boundary of the central square.](image2)
In a discrete system, this yields the update rule

$$E(i, j, d) \leftarrow (1 - 4\lambda) E(i, j, d) + \lambda \sum_{(k, l) \in \mathcal{N}_4} E(i + k, j + l, d),$$

(5)

where \(\mathcal{N}_4 = \{(-1, 0), (1, 0), (0, -1), (0, 1)\}\) is the local neighborhood containing the four direct neighbors.

Equation (5) defines an iterative algorithm for diffusing support, given the initial condition \(E = E_0\). The value of \(\lambda\) controls the speed of the diffusion. At each iteration a new value of \(E\) is computed at every point in disparity space from the current values of the point’s immediate neighbors. A value of \(\lambda < 0.25\) is needed to ensure convergence; we use \(\lambda = 0.15\) for the experiments reported in this paper.

Aggregation using a finite number of simple diffusion steps yields fairly similar results to using square windows. Advantages include the rotational symmetry of the support kernel and the fact that points further away have gradually less influence. The problem of co-opting corners still exists, however.

4.1. Membrane model

A problem with simple diffusion is that the size of the support region increases with the number of iterations. In other words, while the diffusion would eventually converge to a uniform support covering the whole image, we are interested in an intermediate time step in which the diffusion has only progressed to a certain amount. We can change this behavior by adding a term to the diffusion equation that measures the amount each current value has diverged from its original value, yielding the membrane equation (Terzopoulos 1986; Szeliski & Hinton 1985).

$$\frac{\partial E}{\partial t} = \nabla^2 E + \beta (E_0 - E).$$

(6)

In the discrete implementation we use

$$E(i, j, d) \leftarrow [1 - \lambda(\beta + 4)] E(i, j, d) + \lambda \beta E_0(i, j, d) + \sum_{(k, l) \in \mathcal{N}_4} E(i + k, j + l, d),$$

(7)

Unless noted otherwise, we use the parameters \(\lambda = 0.15\) and \(\beta = 0.5\) in the experimental results shown in this paper. The \(\beta\)-term ensures that the diffusion converges to a stable solution not too far from the original values. A closed-form solution for the support function can easily be derived using Fourier analysis (see Appendix A).

Figure 7 shows the results of applying our diffusion process to the \(rds\) image pair. The amount of support at each discrete disparity level is shown before diffusion \((E_0)\), after one iteration, and after 10 iterations. Dark regions indicate strong support for a match. Figure 8 shows the results for accumulating support using the membrane model for the ramp and \(rds\) pairs. The number of diffusion iterations is \(n = 10\) (the results are almost identical at \(n = 5\)).

Using the membrane model alleviates the contour blurring problem to some extent, since the \(\beta\)-term “ties” the center of each support region to its original value. For very noisy images, however, \(\beta\) needs to be chosen quite small to enable enough smoothing for stable matching, making the process more similar to regular diffusion.

4.2. Diffusion with local stopping criteria

A different strategy for preventing both corner co-opting and diffusion to uniformity is to locally stop the diffusion process depending on the distribution of values in each disparity column. To do this, we associate a measure of certainty \(C(i, j)\) with each location. Intuitively, this measure should reflect how “clear” a minimum there is among the values \(E(i, j, d)\) for all \(d\). Given such a measure \(C\), we can aggregate support using non-uniform diffusion:

For each \((i, j)\), compute certainties \(C\) and \(C'\) before and after a single iteration of diffusion. If \(C > C'\), do not diffuse, i.e., restore the old values \(E(i, j, d)\) for all \(d\).

The idea is that diffusion takes place only at locations of ambiguous matches. Also, certainties never decrease, thus guaranteeing convergence.

We have experimented with several different certainty measures. In this paper we will discuss two measures, the winner margin, and the entropy. The winner margin \(C_w\) is the normalized
Disparity level: 0 1 2 3 4 5 6 7 8

\[
\begin{array}{|c|c|c|c|c|c|c|c|c|}
\hline
E_0 & & & & & & & & \\
\hline
E_n = 1 & & & & & & & & \\
\hline
E_n = 10 & & & & & & & & \\
\hline
\end{array}
\]

Fig. 7. Sections through the disparity space of the \textit{rds} image pair during diffusion using the membrane model. The initial disparity space \(E_0\) is displayed at the top. The diffused disparity space \(E\) is shown after one iteration (middle) and after 10 iterations (bottom). Dark regions indicate a match.

difference between the minimum \(E_{\text{min}}\) and the second minimum \(E_{\text{min}}^2\) in a disparity column:

\[
C_m(i, j) = \frac{E_{\text{min}}^2 - E_{\text{min}}}{\sum_d E(i, j, d)}.
\]  

(8)
The second measure \(C_e\) is the negative entropy of the probability distribution in the disparity column. We convert to probabilities by taking the

\[
C_e(i, j) = -\sum_d p(d) \log p(d),
\]

with \(p(d) = \frac{e^{-E(i, j, d)}}{\sum_d e^{-E(i, j, d)}}\).

We will develop the idea of converting to probabilities further in the next section.

Figure 9 shows disparity maps for the \textit{ramp} pair computed with four kinds of diffusion and increasing iterations. The first row shows regular diffusion, the second and third row show diffusion with local stopping based on \(C_m\) and \(C_e\). The fourth row shows diffusion using the membrane model for comparison. It is clearly visible that regular diffusion keeps blurring the features as the number of iteration increases, while the other three diffusion processes converge quickly to a stable solution. Which of the three performs best is hard to tell by looking at the disparity maps; a quantitative analysis based on errors in the computed disparities will be presented in Section 6. It can be seen, however, that none of the diffusion methods does a very good job at recovering the occlusion boundaries. We now turn to a different diffusion algorithm, derived from a Bayesian model of stereo matching, which will result in markedly improved performance.
5. A Bayesian model of stereo matching

Many stereo matching algorithms can be interpreted as approximations to an optimal Bayesian estimator. In this section, we develop a Bayesian model for stereo matching that includes both a measurement model corresponding to the matching criterion and a prior Markov Random Field model corresponding to the aggregation function. Our model uses robust (non-Gaussian) statistics to handle gross errors and discontinuities in the surface. We also develop a novel approximation algorithm that results in a non-linear diffusion process, and show how this produces better results than standard diffusion.

As before, stereo reconstruction is specified as the estimation of a discrete disparity field $d_{i,j} = d(x_i, y_j)$ given two (or more) input images $I_L(x, y)$ and $I_R(x, y)$. Using a Bayesian framework, we first specify a model of image formation, and then derive estimation algorithms from this model.

5.1. The prior model

The Bayesian model of stereo image formation consists of two parts. The first part, a prior model for the disparity surface, uses a traditional Markov Random Field (MRF) to encode preferences for smooth surfaces (Geman & Geman 1984). This model is specified as a Gibbs distribution $p_P$, the exponential of a potential function $E_P$:

$$ p_P(d) = \frac{1}{Z_P} \exp(-E_P(d)), \quad (10) $$

where $d$ is the vector of all disparities $d_{i,j}$ and $Z_P$ is a normalizing factor. The potential function itself is the sum of clique potentials

$$ E_P(d) = \sum_{c \in C} E_c(d) $$

which only involve neighboring sites in the field. In this paper, we study only first order fields, where

$$ E_P(d) = \sum_{i,j} \rho_P(d_{i+1,j} - d_{i,j}) + \rho_P(d_{i,j+1} - d_{i,j}) \quad (11) $$

(see (Terzopoulos 1986; Szeliski 1989) for generalizations to higher order fields).

When $\rho(x)$ is a quadratic, $\rho(x) = x^2$, the field is a Gauss-MRF, and corresponds in a probabilistic sense to a first order regularized (membrane) surface model (Terzopoulos 1986; Szeliski 1989). When $\rho(x)$ is a unit impulse, $\rho(x) = 1 - \delta(x)$, it corresponds to a MRF that favors fronto-parallel
surfaces (Geman & Geman 1984; Marroquin et al. 1987). In between these two extremes are functions derived from robust statistics (Huber 1981), which behave much like surface models with discontinuities (Blake & Zisserman 1987; Geiger & Giroi 1991; Black & Rangarajan 1996). A wide variety of robust penalty functions are possible. In this paper, we use a contaminated Gaussian model,

$$\rho_P(x) = -\log \left( (1-\epsilon_P) e^{-\frac{x^2}{2\sigma_P^2}} + \epsilon_P \right).$$

Figure 10 shows the shape of this function for $\epsilon_P = 0.01$ and $\sigma_P = 1$.

Black & Rangarajan (1996) discuss the relationship between robust penalty methods and nonlinear diffusion techniques such as those of Perona and Malik (1990).

5.2. The measurement model

The second part of our Bayesian model is the data or measurement model which accounts for differences in intensities between left and right images. This model assumes independent, identically distributed measurement errors,

$$p_M(I_L, I_R | d) = \prod_{i,j} p_M(I_L(x_i + d_{i,j}, y_j) - I_R(x_i, y_j)).$$

As mentioned before, traditional stereo matching methods use either a squared intensity error metric (Gaussian noise), $\rho_M(x) = \log p_M(x) = x^2$, or an exact binary matching criterion (e.g., for random-dot stereograms or binary features such as edges or the sign of the Laplacian), $\rho_M(x) = 1 - \delta(x)$. Here we again use a contaminated Gaussian model,

$$\rho_M(x) = -\log \left( (1-\epsilon_M) e^{-\frac{x^2}{2\sigma_M^2}} + \epsilon_M \right),$$

for model Gaussian noise and allow possible outliers due to occlusions or non-modeled photometric effects such as specularities.

The posterior distribution, $p(d|I_L, I_R)$ can be derived from the prior and measurement models using Bayes’ rule,

$$p(d|I_L, I_R) \propto p_P(d) p_M(I_L, I_R | d).$$

As is often the case, it is more convenient to study the negative log probability distribution

$$E(d) = -\log p(d|I_L, I_R)$$

$$= \sum_{i,j} \rho_P(d_{i+1,j} - d_{i,j}) + \rho_P(d_{i,j+1} - d_{i,j})$$

$$+ \sum_{i,j} \rho_M(I_L(x_i + d_{i,j}, y_j) - I_R(x_i, y_j)).$$

While $p(d|I_L, I_R)$ specifies a complete distribution, usually only a single optimal estimate of $d(x, y)$ is desired (but see Szeliski 1989 why modeling of uncertainties may be useful). The most commonly studied estimate is the peak of the distribution, or Maximum A Posteriori (MAP) estimate, which is equivalent to minimizing the energy given in (16). Alternative estimates include quantities such as the mean of the distribution (Marroquin et al. 1987).

A variety of techniques have been developed for minimizing equations like (16). Two of the most popular are the Gibbs Sampler (Geman & Geman 1984; Marroquin et al. 1987) and mean field theory (Geiger & Giroi 1991; Zerubia & Chepalla 1993). The Gibbs Sampler randomly chooses values for each $d_{i,j}$ site according to the local distribution determined by the current guesses for a site’s neighbors (Geman & Geman 1984; Szeliski & Hinton 1985; Barnard 1989). This process will in theory converge to a statistically optimal sample, given enough time. Mean field theory updates an estimate of the mean value of $d_{i,j}$ at each site using a deterministic update rule derived from the original probability distribution (Geman & Geman 1984). It is not guaranteed to find an optimal estimate, but in practice it often finds a good solution, similar to one available through continuation methods (Blake & Zisserman 1987).
5.3. Explicit local distribution model

The Gibbs Sampler and its variants can produce good solutions, but at the cost of long computation times. Mean field techniques, on the other hand, are not very good at modeling ambiguous estimates, such as multiple potential matches at each pixel. Instead of using either of these two traditional approaches, we will develop a novel estimation algorithm based on modeling the probability distribution of \( d_{ij} \) at each site. To do this, we associate a scalar value between 0 and 1 with each possible discrete value of \( d \) at each pixel \((i,j)\), and require that

\[
\sum_d p(i,j,d) = 1. \tag{17}
\]

Our representation is therefore the same as that used by diffusion-based algorithms, i.e., we explicitly model all possible disparities at each pixel, rather than modeling a single estimated disparity as in traditional Gibbs Sampler or mean-field approaches (Barnard 1989).

To initialize our algorithm, we calculate the probability distribution for each pixel \((i,j)\) based on the intensity errors between matching pixels, i.e.,

\[
p_0(i,j,d) \propto \exp(-E_0(i,j,d)), \tag{18}
\]

where

\[
E_0(i,j,d) = \rho_M(I_L(x_i + d, y_j) - I_R(x_i, y_j)) \tag{19}
\]

is the matching cost of pixel \((i,j)\) at disparity \(d\). This is equivalent to a Maximum Likelihood estimate of the probability \(p_0\) given the initial per-pixel matching cost \(E_0\) (without taking into account the spatial prior \(p_P(d)\)).

To derive the update formula, we start with a basic observation about Markov Random Fields: if the joint probability distribution of all interacting neighbors is known, the local probability distribution of a site is completely determined. To compute this distribution, we take the part of the potential energy (16) which involves \((i,j)\), i.e.,

\[
\bar{E}(d_{ij}||d_{i+k,j+l}) = E_0(i,j,d) + \sum_{(k,l) \in N_4} \rho_P(d_{i+k,j+l} - d_{ij}), \tag{20}
\]

and turn this into a probability distribution

\[
p_0(i,j,d) \prod_{(k,l) \in N_4} \sum_{d_{i+k,j+l}} \exp(-\rho_P(d_{i+k,j+l} - d_{ij})) \tag{21}
\]

We then integrate out all of the neighboring disparities according to their joint probability distribution

\[
p(d_{ij}) \propto \sum_{d_{i+k,j+l}} \bar{p}(d_{ij}||d_{i+k,j+l}) p(d_{i+k,j+l}). \tag{22}
\]

In practice, however, it is impossible to estimate the full joint probability distribution of the neighbors, without resorting to a statistical technique such as the Gibbs Sampler. (This is not true, however, of 1-D processes such as Markov Chains.) Instead, we assume (sub-optimally) that the neighboring disparity columns have independent distributions

\[
p(d_{i+k,j+l}) = \prod_{(k,l) \in N_4} p(d_{i+k,j+l}) \tag{23}
\]

where the \(p(d_{i+k,j+l})\) are the current probability density estimates for each neighboring site \((i+k,j+l)\). This assumption resembles the pseudo-likelihood assumption used in the Iterated Conditional Mode (ICM) algorithm (Besag 1986; Zhang et al. 1994). However, ICM further assumes that the individual probability distributions are represented by the mode, i.e., the MAP estimate, whereas we model a complete distribution \(p(d_{i+k,j+l})\). The assumption that neighboring probability distributions are independent is called a factored probability approximation, and is often used to generate mean-field approximation to physical systems. Appendix B shows how our update formula can be derived as the rule which minimizes the Kullback-Leibler divergence between the true posterior Gibbs distribution and its factored (mean-field) approximation.

The complete update formula is

\[
p(d_{ij}) \propto p_0(i,j,d) \prod_{(k,l) \in N_4} \left[ \sum_{d_{i+k,j+l}} \exp(-\rho_P(d_{i+k,j+l} - d_{ij}) p(d_{i+k,j+l}) \right] \tag{24}
\]
or

\[ E(i, j, d) \leftarrow E_0(i, j, d) + \sum_{(k,l) \in \mathcal{N}_d} \log \left[ -\sum_{d'} \exp \left( -\rho_P(d' - d) - E(i+k, j+l, d') \right) \right], \quad (25) \]

where we have replaced \( d_i+k,j+l \) with \( d' \).

For notational and computational convenience, we will introduce a few more additional quantities. The \textit{smoothed probability distribution}

\[ p_s(i, j, d) = \sum_{d'} e^{-\rho_P(d' - d)} p(i, j, d') \]

\[ = \sum_{d'} w_P(d' - d) p(i, j, d') \]

is simply the current probability distribution \( p(i, j, d) \) after it has been convolved \textit{vertically} (in disparity) with the smoothing kernel \( w_P(d) \propto e^{-\rho_P(d)} \), with \( \sum_{d} w_P(d) = 1 \). It has a corresponding \textit{smoothed energy}

\[ E_S(i, j, d) = -\log p_s(i, j, d). \quad (27) \]

Finally, the update rule can be written as a pair of equations

\[ E(i, j, d) \leftarrow E_0(i, j, d) + \sum_{(k,l) \in \mathcal{N}_d} E_S(i+k, j+l, d), \quad (28) \]

\[ p(i, j, d) \leftarrow \frac{e^{-E(i, j, d)}}{\sum_{d'} e^{-E(i, j, d')}} \quad (29) \]

In practice, it useful to introduce an extra parameter \( \mu \) that controls the speed of the diffusion (similar to \( \lambda \) in Equations 5 and 7), and to include the current estimated energy in the update rule. This yields a modified version of (28)

\[ E(i, j, d) \leftarrow E_0(i, j, d) + \mu \left[ E_S(i, j, d) + \sum_{(k,l) \in \mathcal{N}_d} E_S(i+k, j+l, d) \right]. \quad (30) \]

A value of \( \mu < 1 \) slows the diffusion process and facilitates stable convergence (we use \( \mu = 0.5 \)).

If we interpret the above Equations (26), (27), (30), and (29) as a four-step algorithm for iteratively computing the best stereo matches, we see that they are a special instance of a non-linear diffusion process. This is illustrated in Figure 11.

The smoothing step (Equations (26), (27)) blurs the current disparity probabilities vertically along a column, thereby enabling different nearby disparities to support each other (depending on the size of \( \sigma_P \)). It also adds a small amount to each probability \( (\epsilon_P) \), which in effect limits the largest possible value that \( E_S \) can take and thus limits the effect of disparity discontinuities. This step also qualitatively resembles the local winner take all (WTA) step of Yang et al. (1993), in that neighboring disparities are used during the support aggregation stage.

The update step (Equations (30), (29)) is identical to a regular diffusion step with \( \beta \)-terms (membrane model). However, the probability renormalization step ensures that the energies represent meaningful log probabilities (in practice, it forces the smallest \( E \) to be slightly above 0). The robust form of the \( E_S \) function also ensures that bad matches have only limited effects, thus allowing for occlusions or other non-modeled errors to occur.

For the above algorithm to work well, the various parameters \( \{ \sigma_P, \epsilon_P, \sigma_M, \epsilon_M \} \) must be set to appropriate values. \( \sigma_M \) and \( \epsilon_M \) are based on the expected noise in the image sensor, i.e., \( \sigma_M \) should be proportional to the regular image noise, while \( \epsilon_M \) should be the probability of gross errors or oc-
clusions (say 1–10%). The choice of $\sigma_P$ depends on the class of disparity surfaces which may be expected, i.e., a small $\sigma_P$ favors fronto-parallel surfaces. For the experiments presented in this paper, we set $\sigma_P = 0.1$ and $\epsilon_P = 0.01$.

Figure 12 shows the results of our probabilistic aggregation technique applied to the ramp and rds images. We use a different $\sigma_M$ for the two image pairs: $\sigma_M = 2$ for ramp; $\sigma_M = 20$ for rds, to compensate for the different signal strengths of the two pairs. The other parameters are the same for both image pairs: $\epsilon_M = 0.1, \sigma_P = 0.1, \epsilon_P = 0.01$. The number of diffusion iterations is $n = 10$.

6. Experimental results

In this section we numerically evaluate the performance of the different algorithms on synthetic images. We also show results for real image data.

For our experiments we use five synthetic image pairs, based on combining three different intensity patterns ramp, rds, and real, and two different disparity patterns, square and bars. We have already introduced the square disparity pattern (Figure 3), and the combinations ramp/square and rds/square (Figure 4).

The new disparity pattern bars consists of two rectangular regions with two different disparities (see Figure 13). The narrow region in the bottom half of the image is displaced by more than twice its width, thus violating the commonly assumed monotonicity (ordering) constraint. Together with the big disparity range, this provides an extra challenge to stereo algorithms, but reflects common situations in real images. The new

![Figure 12](image1.png)

Fig. 12. Performance of the probabilistic model on the ramp and rds image pairs (gray level images and isometric plots).

![Figure 13](image2.png)

Fig. 13. The bars disparity pattern, containing an ordering constraint violation: (a) isometric plot; (b) gray-level encoding; (c) gray-level encoding with occlusion information.

![Figure 14](image3.png)

Fig. 14. The three additional synthetic image pairs.
Figure 14 shows the three new image pairs synthesized using the texture/disparity combinations real/square, rds/bars, and real/bars. We do not use the combination ramp/bars since the narrow region cannot be matched unambiguously, resulting in meaningless disparity error statistics. All images have size $64 \times 64$; the tested disparity ranges are $0-8$ (square) and $0-27$ (bars).

We compared the following algorithms: SSD, diffusion using the membrane model, diffusion with local stopping, and diffusion using the probabilistic model. For each algorithm, we varied the parameters: window size (SSD), $\beta, \lambda$ (membrane), certainty measure (local stopping), $\sigma_M, \sigma_P, \epsilon_M, \epsilon_P, \mu$ (probabilistic), and the number of iterations (all diffusion algorithms). For each parameter setting, we ran the algorithm on a test set of 40 images (the 5 image pairs with 8 different levels of additive Gaussian noise: $\sigma = 0, 0.25, 0.5, 1, 2, 4, 8, 10$). We tried more than 70 different parameter settings, resulting in about 3000 experiments. In each experiment, we compared the computed disparities with the true disparities (ignoring the occluded regions), and collected three different error statistics: mean absolute disparity error, root-mean-square (RMS) disparity error, and the "percentage of bad points", i.e., the percentage of points whose absolute disparity error is greater than $1/2$.

Recall that our goal in devising the different algorithms was to recover the occlusion boundaries correctly. The percentage of bad points gives a good indication whether the boundaries are recovered correctly, since this is where the errors are big. For similar reasons, we prefer the RMS error over the mean absolute error since it penalizes outliers more.

First we analyzed the error statistics for each method separately to gain understanding of the effect of the different parameters. Then we chose the best parameters for each method, and compared the different methods with each other. We present in detail the results of the second, comparative stage, after briefly discussing the general trends we noticed.

SSD, which we include for comparison, has only one parameter: the size of the support region. The same holds for simple diffusion, where the size of the support region is controlled by the number of iterations. Not surprisingly, the optimal size of the support region depends on the noise level. In general, higher noise levels (or, more precisely, lower signal-to-noise ratios) require bigger window sizes. The best window size can also depend on the image.

The membrane model behaves similarly to regular diffusion with a fixed number of iterations. For small noise levels, a value of $\beta$ between $1/3$ and 1 usually yields smaller errors than regular diffusion, but not always. Also, as mentioned before, for high noise levels, $\beta$ needs to be chosen quite small to enable enough smoothing for stable matching.

In analyzing regular diffusion with local stopping criteria, we found that the certainty measure is critical. In our experiments, the winner margin $C_m$ almost always outperformed the measure based on entropy $C_e$. A problem with our definition of local stopping is that an initial wrong but "certain" match can survive. There is clearly a potential for both better certainty measures and different stopping criteria.

The probabilistic model, which performed by far the best, also has the most parameters. We found, however, that many parameters have only small effects and can be set to default values, including $\epsilon_M = 0.1, \epsilon_P = 0.01$, and $\mu = 0.5$. As expected, a small $\sigma_P$ worked best for our test images composed from fronto-parallel surfaces. For real images, we found that $\sigma_M$ needs to be chosen slightly higher. The most important parameter is $\sigma_M$, which should reflect the strength of the image signal. We used three different values for the three different textures of our test images. Finally, the number of iterations is less critical, since the method seems to converge relatively fast to a stable solution. Higher numbers of iterations are necessary for images containing regions of uniform intensity, such as the real images discussed below.

For direct comparison of the methods, we plot the disparity error versus the noise level on all five image pairs: Figure 15 shows the RMS errors, and the percentage of bad points. We compare SSD with a window size of 5, the membrane model with $\beta = 0.5$, diffusion with local stopping based on winner margin $C_m$, and the probabilistic model with $\epsilon_P = 0.01, \sigma_P = 0.1, \epsilon_M = 0.1$, and
Fig. 15. Comparative performance of four stereo algorithms on five test image pairs. The plots show two different performance measures as a function of the standard deviation of image noise. The left column shows the RMS error of the computed disparities; the right column shows the percentage of points whose absolute disparity error is greater than $1/2$. Disparities of occluded points are not included.
The probabilistic model clearly beats the three other methods. For small noise levels, the occlusion boundaries are recovered almost perfectly (the percentage of bad points is 0% in three of five images). Note that the algorithm recovers the "correct" disparity pattern, even though the notion of true disparities is not well defined for ambiguous images such as random dot stereograms.

We also tested our algorithms on real images. We include results of the probabilistic method on images from the SRI’s tree sequence and CMU’s town sequence (see Figure 16). We used multiple baseline stereo based on five images to initialize the disparity space with the sum of four (appropriately scaled) similarity measures (Okutomi & Kanade 1993). Figure 17 shows the disparity maps computed by the probabilistic algorithm after 50 iterations, using the following parameters: \( \sigma_P = 0.4, \epsilon_P = 0.01, \sigma_M = 5, \epsilon_M = 0.1 \). Note that we use a bigger \( \sigma_P \) than before to account for slanted surfaces.

The running times are 220 seconds for the tree pair (image size: \( 256 \times 233 \), disparity levels: 16), and 119 seconds for the town pair (image size: \( 240 \times 256 \), disparity levels: 9). Thus, on average about 4.5 microseconds are spent per pixel per disparity per iteration. These times were obtained on a DEC Alpha workstation using an experimental (sequential) implementation that was not optimized for speed.

7. Discussion

As we have shown, linear and non-linear diffusion algorithms are an attractive alternative to the adaptive windows introduced by Kanade and Okutomi (1994). In its simplest form, the membrane algorithm simply requires the iterative summation of neighboring matching costs, with an additional term thrown in to prevent the support region from growing indefinitely. The increased weighting of the central pixel relative to the periphery is sufficient to counteract many of the artifacts introduced by the squared summation window used in SSD. When combined with a local stopping criterion, the resulting non-linear diffusion process has an adaptive support behavior similar to the variable window size algorithm. The inclusion of additional non-linearities in the Bayesian diffusion algorithm improves the performance even more. The Bayesian formulation has the property that the prior term dominates in regions without much texture, and the data term dominates in regions with great texture variations. Thus, the non-linear diffusion algorithm derived from the Bayesian formulation also has an implicit adaptive windowing effect. In addition, the accurate recovery of the occlusion boundaries is aided by using robust penalty functions both for the smoothness prior and for computing the matching cost.

In addition to their simplicity and computational efficiency, our non-linear diffusion algorithms can also handle stereograms with more ambiguity than the adaptive window SSD algorithm. Kanade and Okutomi’s algorithm is based on locally adjusting the sub-pixel disparity estimate simultaneously with growing the window size. This presupposes that the algorithm is somehow initialized in the vicinity of the true disparity. This is achieved in their synthetic image sequences by using small disparities, and in their real sequences by using a multi-frame version of the basic SSD algorithm (Okutomi & Kanade 1993). Image pairs with rapidly varying textures and many potential matches such as the random-dot stereograms used in our experiments could not be handled by their current algorithm. Of course, their basic method could potentially be extended to include a standard multiple disparity search component, but the performance of such a hybrid method is as yet unknown.

In its present form, our algorithm computes monocular rather than binocular disparity maps, i.e., the disparity map is associated with the right image. A binocular representation would remove this restriction, enabling the representation of occluded regions in both left and right images. Extending our diffusion algorithms to a binocular representation is relatively straightforward: the concept of neighbors at the same disparity is modified to define equal disparities in the cyclopean representation of depth, i.e., the depth seen by a camera halfway between the original two (Barnard 1989). Such a representation would also allow us to deal with occlusions more gracefully, allowing
occluded pixels to float to the same disparity as other pixels in the background. However, it is unclear how to extend the Bayesian algorithm, since it requires the re-normalization of disparities along each column in disparity space.

An alternative strategy for handling occlusions is to explicitly prevent pixels that have already been assigned to a more frontal surface from participating in the matching cost evaluation in regions that are “shadowed” by the frontal surface (Szeliski & Golland, 1998). The prior smoothness model would then cause these regions to be filled in at the background depth (and not at the frontal depth, where the matching cost would not have been suppressed). In addition to these extensions, we also plan to study better local stopping criteria based on improved certainty measures. We would also like to investigate multi-resolution versions of our diffusion algorithms to help fill in regions which have few features to match. One possibility would be to use the multi-resolution framework of Yang et al. (1993).

8. Conclusions

In this paper, we have demonstrated that diffusion-based aggregation of support is a useful alternative to both traditional area-based correlation and to more recent adaptive window size-based techniques. Our algorithms are simple to implement and computationally efficient, and result in better quality estimates, especially near
discontinuities in the disparity surface. The addition of local termination conditions to the basic diffusion process results in a behavior similar to that of adaptively sized windows. Furthermore, our novel non-linear diffusion algorithm derived from a Bayesian model of stereo matching results in markedly improved performance. We believe that further study of the basic support and aggregation methods in stereo matching is central to developing algorithms with improved performance over a wide range of imagery.

Acknowledgements

We would like to thank the anonymous reviewers for their helpful comments. Special thanks to Chris Williams for pointing out the relation of our Bayesian algorithm with the Potts Glass model, and for his help on Appendix B.

A portion of this work was performed while Richard Szeliski was employed at Digital Equipment Corporation's Cambridge Research Lab.

Appendix A

Support function for the membrane model

The support function (i.e., impulse response or kernel) for the membrane diffusion model is a function which can be convolved with the original input data $E_0$ to yield the final value of $E$. This function can be computed by setting $E_0$ to a unit impulse $E(i,j)=\delta(i)\delta(j)$, and setting the r.h.s. of (6) to 0.

For the discrete case (7), this involves solving the coupled set of equations

$$\beta (\delta(i)\delta(j) - f(i,j)) + \sum_{(k,l)\in \mathcal{N}_4} (f(i+k,j+l) - f(i,j)) = 0$$

(A1)

(the support function is the same for all disparity levels $d$). Re-writing these in the Fourier domain, we obtain

$$\beta (1 - F(\omega_x,\omega_y)) + \sum_{(k,l)\in \mathcal{N}_4} \left( F(\omega_x,\omega_y)e^{j(\omega_xk + \omega_yl)} - F(\omega_x,\omega_y) \right) = 0$$

or

$$F(\omega_x,\omega_y) = \frac{\beta}{\beta + 4 - 2\cos \omega_x - 2\cos \omega_y}. \quad (A2)$$

While the inverse transform of $F(\omega_x,\omega_y)$ has no closed form solution, it is simple enough to compute numerically (see Figure A.1 for a plot).

![Shape of the membrane support function for $\beta = 0.5$](image)

**Fig. A.1.** Shape of the membrane support function for $\beta = 0.5$

Appendix B

Mean field analysis of a Potts glass

Assume we have a posterior probability distribution over $d_i$ which is a Gibbs distribution, so that

$$-\log P(d|I_L, I_R) = \sum_{i,j} E_{ij}(d_i,d_j) + \sum_i E_i(d_i) + \log Z$$

(B1)

where each $d_i$ can only take on a discrete set of values 1..K and $Z$ is the partition function. This kind of a model is called a Potts glass (Peterson & Söderberg, 1989).

Relating this back to our Bayesian stereo matching model (Section 5), the index $i$ refers to a pixel $(i,j)$, the $d_i$ are the disparities; $E_{ij}$ and $E_i$ encode the prior and measurement models, $E_P(d)$ and $E_M(I_L, I_R|d)$.

Now, represent each $d_i$ by its distributed representation $s_i$, which is a $K$-dimensional bit vector, with only one bit being on at a time (the $k$th bit
being on is equivalent to \( d_i = k \).

We can then rewrite the log likelihood of the Gibbs distribution as

\[
-\log P(d|l_L, l_R) = \sum_{ij} s_i^T A_{ij} s_j + \sum_i b_i^T s_i,
\]

where the \( kl \)-th entry in \( A_{ij} \) is \( E_{ij}(d_i = k, d_j = l) \) and the \( k \)-th entry in \( b_i \) is \( E_{ik}(d_i = k) \).

Computing either the ground state of the Gibbs distribution (i.e., the MAP estimate) or even the marginal distribution of a given variable \( d_i \) is generally computationally intractable, although one can try to use approximate algorithms, e.g., the Gibbs Sampler.

As an alternative, we can choose to approximate the original probability function \( P(d|l_L, l_R) \) with a simpler factored distribution

\[
Q(d) = \prod_i Q_i(d_i)
\]

i.e., we assume that the joint probability distribution is a product of independent per-site probabilities. This approximation is often called the mean-field approximation (Parisi, 1988).

In order to find the best mean-field approximation, we minimize the Kullback-Leibler divergence between the new and original distributions (Cover & Thomas, 1991)

\[
D_{KL} = \sum_d Q(d) \log \frac{Q(d)}{P(d)}.
\]

where the summation is taken over all states. This can be written as

\[
D_{KL} = H(Q) - \sum_d Q(d) \log P(d),
\]

where \( H(Q) \) is the entropy of the distribution \( Q \). As \( Q(d) \) is factored, \( H(Q) \) is given by

\[
H(Q) = \left< \log \prod_i Q_i(d_i) \right> = \sum_i \log Q_i(d_i) = \sum_i \sum_k q_{ik} \log q_{ik},
\]

where \( q_{ik} = Q_i(d_i = k) \) is the (marginal) probability that \( d_i \) is equal to \( k \), or equivalently, the expected or mean value of the \( k \)-th bit of \( s_i \).

The second term in \( (B4) \) can be expanded using \( (B2) \) as

\[
-\sum_d Q(d) \log P(d)
\]

\[
= \left< \sum_{ij} s_i^T A_{ij} s_j + \sum_i b_i^T s_i \right>
\]

\[
= \sum_{ij} q_{ij}^T A_{ij} q_j + \sum_i b_i^T q_i,
\]

where \( q_i = \langle s_i \rangle \) is the mean (expected) value of \( s_i \) (i.e., \( q_{ik} \) is the \( k \)-th component of \( q_i \)).

We wish to minimize the KL divergence subject to the condition that the marginal probabilities sum up to 1, i.e., \( \sum_k q_{ik} = 1 \). We can do this using Lagrange multipliers, i.e., minimizing

\[
C = D_{KL} + \sum_i \lambda_i (\sum_k q_{ik} - 1).
\]

Taking the partial derivative w.r.t. \( q_{ik} \), we get

\[
\frac{\partial C}{\partial q_{ik}} = 1 + \log q_{ik} + \sum_j a_{ij}^k q_{jk} + b_{ik} + \lambda_i = 0,
\]

where \( a_{ij}^k \) is the \( k \)-th column of \( A_{ij} \). We can thus compute the required formula for \( q_{ik} \) in terms of the other \( q_j \) as

\[
q_{ik} = \exp(-\sum_j a_{ij}^k q_{jk} + b_{ik}) \exp(-\lambda_i + 1).
\]

Imposing the requirement that \( \sum_k q_{ik} = 1 \), we can solve for \( \exp(\lambda_i + 1) = \sum_k \exp(-\sum_j a_{ij}^k q_{jk} + b_{ik}) \) to get the final result

\[
q_{ik} = \frac{\exp(-\sum_j a_{ij}^k q_{jk} + b_{ik})}{\sum_k \exp(-\sum_j a_{ij}^k q_{jk} + b_{ik})}
\]

This formula is the direct analogue of the probability updating rule (equation 24) derived in Section 5, after equating the terms

\[
a_{ij}^k = \exp(-p_{d_i+d_j-i})
\]

\[
q_j = p(d_i+d_j-i)
\]

\[
b_{ik} = -\log p_{d_i+d_j-i}^h
\]
Thus, we know that applying this rule to any isolated site will result in a reduction of the Kullback-Leibler divergence, and hence has a guaranteed convergence to at least a local minimum. Applying updates in parallel may not converge, but should work fine if small enough steps are used.

References


Stereo Matching with Non-Linear Diffusion


