ＡＴＲテクニカルレポート表紙

〔非公開〕

ＴＲ－Ｍ－００２６

A New Stereo Matching Algorithm Based on Bayesian Model

ｻﾝﾌｧ ﾘｰ
Sang Hwa Lee

ｼﾞｮﾝｲﾙ ﾊﾟｰｸ 井 上 誠 喜
Jong-Il Park Seiki INOUE

１９９７．８．２９

ＡＴＲ知能映像通信研究所

Technical Report

A New Stereo Matching Algorithm Based on Bayesian Model

Sang Hwa Lee
ATR
Media Integration and Communications(MI&C)
Department 3

Abstract

In this paper, the general formula of Bayesian model for stereo matching algorithm is derived and implemented with simplified probabilistic models. The probabilistic models are independence property between the neighborhood disparities in the configuration, and similarity of disparities in adjacent neighborhood. The formula is the generalization of Bayesian model of stereo matching, and can be changed into the some forms of Bayesian model of stereo matching according to the probabilistic models in the disparity neighborhood system or configuration. And, this paper performed two kinds of experiments. One is the comparison of the performance between the proposed algorithm and the other algorithms, such as the conventional Bayesian algorithm [?, ?] and block-based squared sum algorithm. The other experiment applies the multiple view stereo images to the proposed algorithm of Bayesian model. According to the experimental results, we can conclude the following facts. The first is that the derived formula is the general form and can be changed into the some different forms based on the reasonable probabilistic assumptions. The more accurate is the assumed probability model, the better disparity map can be generated with this formula. And, this Bayesian model can be developed with various probabilistic model and configuration. The second is that it is very important to generate the initial energy space in Bayesian model of stereo matching, so the multiple stereo images are useful for the estimation of the better disparity map.

1 Introduction

Stereo matching is to estimate the disparities between stereo images, which are generated from slightly different viewpoints respectively. Stereo matching generates the three dimensional scene structure, disparity map, from the two dimensional stereo images set. In order to estimate the disparities between stereo images generated from different viewpoints, various algorithms have been proposed. The sum of squared differences(SSD) algorithm searches for the disparity based on the region. This algorithm calculates the squared difference of intensity in stereo images and sums it in the region. Based on this summed difference, it finds the disparity of the minimum summed difference. This algorithm is simple to implement and has an advantage to apply the various post processing in order to reduce the errorneous disparities. However, this algorithm is very dependent on the size of region and suffers from the blurring of the boundary in the disparity map. The gadient based algorithm [?, ?] is based on the idea that the same intensity or color in image may look different corresponding to the visual angles. Due to this misunderstading of human visual system and camera imaging system, the measure based on the intensity or color difference may be incorrect at some angles of viewpoints. This paper proposes the measure of matching as gradient based quantity. This algorithm chooses the disparity with most similar gradient of intensity in the stereo images. In this algorithm, it is dufficult to find the correct disparity in the images to vary uniformly. The diffusion based algorithm is based on the diffusion equation of energy function. That id to say, the gradient of energy function in time domain is equivalent to the Laplacian of the energy function. With this diffusion equation, the energy function is diffused iteratively and the disparity of minimum energy is selected. Since this algorithm is linear function, some of nonlinear adaptivity have been proposed. And, there are many algorithms to be combined with the above indivisual algorithms.
In this paper, we consider the Bayesian model of stereo matching. This algorithm is based on the probabilistic model in order to find the disparity. Given stereo images set, this algorithm searches for the disparity which has maximum probability. Also, a new probability distribution, Gibb's distribution, is introduced to evaluate the probability of disparity, since it is difficult to evaluate the probability directly.

This paper consists of six sections. In section 2, we will survey the basic theory of Bayesian model for stereo matching. The section 3 derives the general formula of Bayesian model for stereo matching, and derives a prctical formula based on the probabilistic assumptions. In the section 4, a proposed algorithm is described. The experimental results are shown in the section 5, and finally, we conclude the proposed algorithm and experimental results in the section 6.

2 Bayesian Model of Stereo Matching

In the stereo matching to estimate the disparity map between the stereo images generated from different viewpoints respectively, the starting point is to maximize the conditional probability of disparity given stereo images.

$$maximize \quad p(\,\mathbf{D} \mid \{\mathbf{I}\}\,), \qquad \{\mathbf{I}\} \;=\; \{\,\mathbf{I}_R,\ \mathbf{I}_K,\ \ K = integer\,\}\,. \tag{1}$$

In *eq.* (1), the symbol $\{\mathbf{I}\}$ is the stereo image set and this set is ususally composed of two images, left and right viewpoint, respectively. $\mathbf{I}_R$ is the reference image to compare with the other images in the stereo image set. The symbol $\mathbf{D}$ is the disparity map of the stereo image set and describes the three dimensional structure between the stereo images. Before maximizing the conditional probability, however, the main problem of stereo matching is how to calculate the conditional probability given only stereo image set. Therefore, we need a new measure to transform an available measure into probability space measure. The Gibb's distribution is the probability distribution whose random variable is related to the energy dimension.

$$p(x) \;=\; C\ \exp\left\{\, -\frac{E(x)}{T} \,\right\}. \tag{2}$$

In the Gibb's distribution, the symbol $E(x)$ is the energy of the random variable x and can be obtained from some functions of x easily. In other words, the Gibb's distribution is a kind of measure to transform the energy space into probability space[?]. Now, we can consider the probability space as energy space with the Gibb's distribution, and, construct the one to one mapping between the two spaces. By transforming *eq.* (1) with the Gibb's distribution such as

$$p(\,\mathbf{D} \mid \{\mathbf{I}\}\,) \;\propto\; \exp\{\,-E(\,\mathbf{D} \mid \{\mathbf{I}\}\,)\,\}\,, \tag{3}$$

we can also define the Bayesian model of stereo matching as another expression.

$$minimize \quad E(\,\mathbf{D} \mid \{\mathbf{I}\}\,). \tag{4}$$

In the energy space, we generally construct measures which are related to the error energy. Therefore, we should minimize the energy in order to maximize the transformed probability. By the Bayesian rule in the probability theory, the conditional probability $p(\,\mathbf{D} \mid \{\mathbf{I}\}\,)$ can be expressed as

$$p(\,\mathbf{D} \mid \{\mathbf{I}\}\,) \;=\; \frac{p(\,\{\mathbf{I}\}\ \mid \mathbf{D}\,)\,p(\mathbf{D})}{p(\,\{\mathbf{I}\}\,)}. \tag{5}$$

Since $p(\,\{\mathbf{I}\}\,)$ is the given and fixed probability, I can consider the only numerator in *eq.* (5),

$$p(\mathbf{D} \mid \{\mathbf{I}\}) \propto p(\{\mathbf{I}\} \mid \mathbf{D}) p(\mathbf{D}). \tag{6}$$

By transforming the probability *eq.* (6) into the energy space based on Gibb's distribution, we can obtain the energy equation which is composed of two energy measures dereived from the intensity based measure model and disparity based similarity model, respectively.

$$E(\mathbf{D} \mid \{\mathbf{I}\}) \propto E(\{\mathbf{I}\} \mid \mathbf{D}) + E(\mathbf{D}) \tag{7}$$

In *eq.* (7), the first term means by the error energy due to the intensity differneces given disparity map, and it can be calculated as follows.

$$E(\{\mathbf{I}\} \mid \mathbf{D}) = \sum_{\mathbf{I}_K \in \{\mathbf{I}\}} \rho_i(\mathbf{I}_R - \mathbf{I}_K(\mathbf{D})). \tag{8}$$

Where, $\mathbf{I}_K(\mathbf{D})$ is the translated $\mathbf{I}_K$ by the disparity map $\mathbf{D}$, and $\rho_i(\cdot)$ is the measure function to construct the energy, and $\rho_i(x) = x^2$ is the usual case. In the case of square function as measure function, however, the Gibb's distribution becomes Gaussian distribution and this is too special case. This paper uses the contaminated Gaussian energy measure which generalizes the energy measure between Gaussian and Dirac Delta distribution.

$$\rho_i(x) = -\log\left((1-\epsilon_i)\exp\left\{-\frac{x^2}{2{\sigma_i}^2}\right\} + \epsilon_i\right) \tag{9}$$

By adjusting the parameter ϵ_i from 0 to 1, we can obtain the distributions between Gaussian and Dirac Delta distribution. If the parameter is zero, the transformed distribution, which results from transformation of energy measure by Gibb's distribution, becomes the Gaussian distribution. Also, if the parameter approach unity, the transformed distribution becomes convergent to the Dirac Delta distribution. And, in *eq.* (7), the second term is the energy measure of disparity map. This measure is based on the assumption that the disparity will distribute continuously in the image plane. That is to say, the disparity will be similar to that of adjacent neighborhood. The energy $E(\mathbf{D})$ increases as the difference between adjacent disparities increases. This energy term has an important role in regularization of disparity ditribution in the diffusion process. The measure function of this energy use the contaminated Gaussian energy measure same as intensity based measure. However, in the case of disparity based measure, the contaminated Gaussian has an important effect on the convergence of diffusion and boundary spreading problem of generated disparity map.

$$\rho_d(x) = -\log\left((1-\epsilon_d)\exp\left\{-\frac{x^2}{2{\sigma_d}^2}\right\} + \epsilon_d\right) \tag{10}$$

In summary of *eq.* (7), there are two error energy measures in the Bayesian model of stereo matching, one is the intensity based error measure and the other is the regularization model based on the assumption of continuous disparity distribution.

In next section, we derive the general equation of the Bayesian model for stereo matching in the image plane. The equation means by the procedure such as calculation of the error energies, transformation into probability, and searching for the maximum probability or minimum error energy.

3 Derivation of the General Formula of Bayesian Model

In this section, we derive the general formula of the Bayesian model for stereo matching in the image plane. Before deriving the general formula at a position (i, j) in the image plane, however, we first need to introduce the concept of energy space in stereo matching. There are two approaches in estimating the disparity from the error measure, point oriented and displacement oriented algorithm. In the point oriented algorithm, the error energy is aggregated in the fixed block over a certain disparity range, and the disparity of minimum error energy is selected. Finally, some post processings are executed in order to reduce the false disparity. The displacement oriented algorithm, on the other hand, calculates the pointwise error energy over the all disparity range, and searches for the minimum energy through the disparity range.

In the step of error calculation, 3 dimensional energy space is constructed. This is shown in Fig. 1. At each point (d, i, j), the error energy is precalculated and it is used and updated iteratively with the derived diffusion equation later. This paper uses the displacement oriented stereo matching algorithm.

3.1 General formulation of Bayesian Model

Let the disparity at the position (i, j) in the image plane be $d_{i,j}$. We consider the pointwise energy $E(d_{i,j})$ in the energy space. And according to Markov Random Field(MRF) theory, it is proved that it is possible to estimate the disparity of a position, if all of the joint distributions between neighborhood disparities are known in advance. In order to calculate the similarity between neighborhood disparity, we have a basis on MRF theory. Therefore, the probability to be maximized becomes the conditional probability given stereo image set $\{\mathbf{I}\}$ and neighborhood disparity set. Let the configuration of neighborhood disparities be $\mathcal{N}$, and disparity vector be

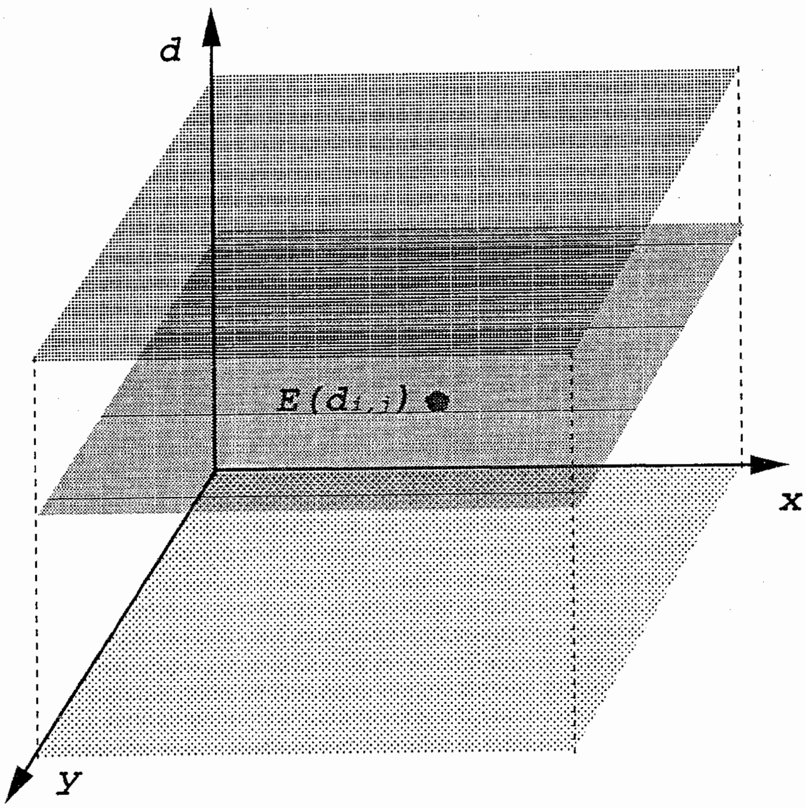


Figure 1: Energy space of displacement oriented approach

$\mathbf{d}_{\mathcal{N}}$, which consists of the disparity elements in the configuration $\mathcal{N}$. The problem of stereo matching based on the MRF theory is as below.

$$minimize \;\; E(\, d_{i,j} \mid \{\mathbf{I}\}\, ,\, \mathbf{d}_{\mathcal{N}}\,), \;\; or, \;\; maximize \;\; p(\, d_{i,j} \mid \{\mathbf{I}\}\, ,\, \mathbf{d}_{\mathcal{N}}\,). \qquad (11)$$

As described above, the energy function to be minimized can be composed of two measures. The first is the intensity based error energy measure given disparity map and the second is the disparity based similarity measure given the configuration $\mathcal{N}$, or, neighborhood disparity vector $\mathbf{d}_{\mathcal{N}}$. One is the error energy measure in the image plane, and the other is the regularization measure of the disparity space. The former is the energy of error signal given the disparity set for stereo matching, and can be characterized by the intensity of the stereo image set and disparity map. The latter is the energy function in order to consider the similarity between neighborhood disparities based on the continuous distribution of disparity in the image plane. The intensity based error $E_0(d_{i,j})$ is calculated with stereo image set $\{\mathbf{I}\}$ and the disparity $d_{i,j}$.

$$E_0(d_{i,j}) \;=\; \frac{1}{n(\{\mathbf{I}\})-1} \sum_{\mathbf{I}_K \in \{\mathbf{I}\}} \rho_i(\, \mathbf{I}_R(i,j) \;-\; \mathbf{I}_K(i, j + d_{i,j})\,) \qquad (12)$$

In this paper and our experiments, we assume the stereo images have a epipolar geometry. So, we have to consider only one component of coordinate system in each pair of images to be compared. The energy function for similarity of disparity, $E(\, d_{i,j} \mid \mathbf{d}_{\mathcal{N}}\,)$ is

$$E(\, d_{i,j} \mid \mathbf{d}_{\mathcal{N}}\,) \;=\; \sum_{d_n \in \mathcal{N}} \rho_d(d_{i,j} - d_n). \qquad (13)$$

With the two measurements equations, *eq.* (12) and *eq.* (13), we can construct the energy function at $d_{i,j}$ in the energy space based on *eq.* (7),

$$E(\, d_{i,j} \mid \{\mathbf{I}\}\, ,\, \mathbf{d}_{\mathcal{N}}\,) \;=\; E_0(d_{i,j}) \;+\; \sum_{d_n \in \mathcal{N}} \rho_d(d_{i,j} - d_n). \qquad (14)$$

In *eq.* (14), since the image set $\{\mathbf{I}\}$ is given and fixed, we can consider only the configuration condition in the probabilistic manipulations. In other words, it is no loss in generality to dealing with the conditional probability, $p(\, d_{i,j} \mid \mathbf{d}_{\mathcal{N}}\,)$. In order to obtain the probability distribution from the energy function, we apply the Gibb's distribution to the energy equation. Since the Gibb's distribution is the probability measure of the energy function, it transforms the energy space into probability space. we rewrite *eq.* (14) as

$$p(\, d_{i,j} \mid \mathbf{d}_{\mathcal{N}}\,) \;=\; p_0(d_{i,j}) \prod_{d_n \in \mathcal{N}} \exp\{\, -\rho_d(d_{i,j} - d_n)\, \}\, . \qquad (15)$$

By the probability theory, the probability of $d_{i,j}$ can be obtained by integrating the above conditional probability over the joint probability distribution of all configurations in the neighborhood system,

$$p(d_{i,j}) = \sum_{all\ \mathbf{d}_{\mathcal{N}}} p(d_{i,j} \mid \mathbf{d}_{\mathcal{N}}) p(\mathbf{d}_{\mathcal{N}}). \tag{16}$$

In *eq.* (16), $p(\mathbf{d}_{\mathcal{N}})$ is the joint probability distribution of disparities in the same configuration. The symbol *all* $\mathbf{d}_{\mathcal{N}}$ means by the all configuarations to be available at $d_{i,j}$. These configurations may be changed corresponding to a position $d_{i,j}$ in the energy space. Let the *all* $\mathbf{d}_{\mathcal{N}}$ define as $\mathcal{S}$, the super set of sets which consists of all configurations of neighborhood disparities with respect to $d_{i,j}$. Also, let $n(\mathcal{S})$ be the number of the elements in the super set $\mathcal{S}$.

$$\mathcal{S} = \{\mathcal{N} \mid \mathcal{N} : \ configuration\ of\ neighborhood\ disparities\ at\ (i,j)\} \tag{17}$$

Combining *eq.* (16) with *eq.* (15),

$$p(d_{i,j}) = p_0(d_{i,j}) \sum_{\mathcal{N} \in \mathcal{S}} \left[p(\mathbf{d}_{\mathcal{N}}) \prod_{d_n \in \mathcal{N}} \exp\left\{-\rho_d(d_{i,j} - d_n)\right\} \right]. \tag{18}$$

The probability space is useful for development of Bayesian model because of the Bayesian model is based on probability theory. However, the probability is unknown in practical situation, and it is also difficult to estimate the probability directly. we always estimate the probability indirectly from any other available measures and transformation. In this paper, energy measure and Gibb's distribution are used for the purpose. *Eq.* (18) can be transformed into energy function as before,

$$E(d_{i,j}) = E_0(d_{i,j}) - \log \left[\sum_{\mathcal{N} \in \mathcal{S}} \left\{ p(\mathbf{d}_{\mathcal{N}}) \prod_{d_n \in \mathcal{N}} \exp\left\{-\rho_d(d_{i,j} - d_n)\right\} \right\} \right] \tag{19}$$

Eq. (19) is the general formula of Bayesian model of stereo matching. With this formula, we can change the energy space iteratively, and find the most accurate disparity after the diffusion is convergent. The intensity based error energy is calculated only once, and the joint distributions of the configurations are needed to diffuse the energy space. The distributions of the configurations have a dominant role in the diffusion process, and estimating the distributions is an important problem in the practical implementations. Now, we manipulate the *eq.* (19) using the inequality between arithmetic mean and geometric mean as follows.

$$\frac{\sum_{k=1}^{K} a_k}{K} \geq \sqrt[K]{\prod_{k=1}^{K} a_k}, \quad for \;\; all \;\; a_k > 0. \tag{20}$$

Since each term of the summation operator in *eq.* (19) is positive, it is possible to apply this inequality to the general formula. we can obtain an inequality by manipulating *eq.* (19) based on *eq.* (20) and, derive another general formula of Bayesian model for stereo matching.

$$\begin{aligned}
& \log \left[\sum_{\mathcal{N} \in \mathcal{S}} \left\{ p(\mathrm{d}_{\mathcal{N}}) \prod_{d_n \in \mathcal{N}} \exp\{-\rho_d(d_{i,j} - d_n)\} \right\} \right] \qquad (21) \\
\geq \; & \log \left[n(\mathcal{S}) \sqrt[n(\mathcal{S})]{\prod_{\mathcal{N} \in \mathcal{S}} \left\{ p(\mathrm{d}_{\mathcal{N}}) \prod_{d_n \in \mathcal{N}} \exp\{-\rho_d(d_{i,j} - d_n)\} \right\}} \right] \\
= \; & \log n(\mathcal{S}) + \frac{1}{n(\mathcal{S})} \left[\sum_{\mathcal{N} \in \mathcal{S}} \left[\log\left(p(\mathrm{d}_{\mathcal{N}}) \right) + \log \left\{ \prod_{d_n \in \mathcal{N}} \exp\{-\rho_d(d_{i,j} - d_n)\} \right\} \right] \right] \\
= \; & \log n(\mathcal{S}) + \frac{1}{n(\mathcal{S})} \left[\sum_{\mathcal{N} \in \mathcal{S}} \left\{ \log\left(p(\mathrm{d}_{\mathcal{N}}) \right) - \sum_{d_n \in \mathcal{N}} \rho_d(d_{i,j} - d_n) \right\} \right],
\end{aligned}$$

In order to calculate the $n(\mathcal{S})$, define the disparity range set, $\mathcal{D}$, as the set of all the possible disparities, and the geometry set of the configurations, $\mathcal{G}$. This set consists of geometric structures of neighborhood in the two dimensional image plane. Fig. 2 shows the various geometries of the neighborhood configurations. As mentioned above, it is possible to vary the geometry of the configuration as well as the disparities in the configuration. The $n(\mathcal{S})$ is calculated as

$$n(\mathcal{S}) = \sum_{k=1}^{n(\mathcal{G})} [n(\mathcal{D})]^{n(\mathcal{N}_k)} , \tag{22}$$

where, $\mathcal{N}_k$ is the configuration with k-th geometry in $\mathcal{G}$. And, combining the *eq.* (19) with *eq.* (20), we can obtain the general formula of Bayesian model for stereo matching as follows.

$$E(d_{i,j}) \leq E_0(d_{i,j}) - \log n(\mathcal{S}) - \frac{1}{n(\mathcal{S})} \left[\sum_{\mathcal{N} \in \mathcal{S}} \left\{ \log\left(p(\mathrm{d}_{\mathcal{N}}) \right) - \sum_{d_n \in \mathcal{N}} \rho_d(d_{i,j} - d_n) \right\} \right] \tag{23}$$

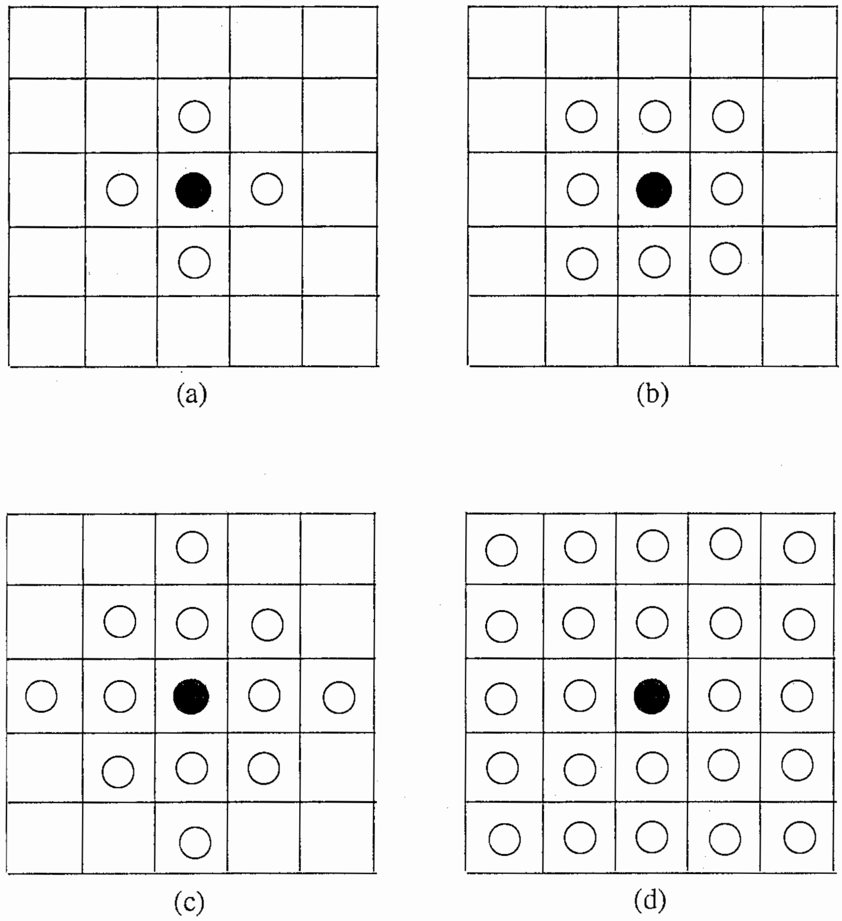


Figure 2: Examples of Geometry of Configuration

The *eq.* (23) shows the upper bound of the energy space at each point in the energy space. In the term $\rho_d(d_{i,j} - d_n)$ of *eq.* (23), as the difference of disparity increases, the upper bound of the energy $E(d_{i,j})$ increases, and the corresponding probability decreases. On the other hand, if the difference decreases, the upper bound of energy decreases and the corresponding probability increases. This means by the regularization of the adjacent disparities. In other words, the formula makes the neighborhood disparities smooth and the isolated disparity be similar to the neighborhood disparities. And, if the term $p(\mathbf{d}_{\mathcal{N}})$ has a large value, the upper bound of energy decreases and the probability increases. Therefore, we should choose the configuration so as to make the joint probability distribution as large as possible. This paper will implement the general formula in the point of how to choose the configuration in order to estimate the joint distribution maximally.

3.2 Independence Probabilistic Model

In the general formula such as *eq.* (19) or *eq.* (23), it is necessary to know the joint distribution $p(\mathbf{d}_{\mathcal{N}})$ of disparities in the same configuration, and to specify the configuration at each $d_{i,j}$ in advance. However, it is difficult to obtain the exact joint probability distribution in real cases. So, we assume the simple probabilistic model for the joint distribution of the configuration. Assume that the disprities in the configuration are independent one another. The joint probability $p(\mathbf{d}_{\mathcal{N}})$ can be rewritten as the product of marginal probabilitie of each disparity in the configuration.

$$p(\mathbf{d}_{\mathcal{N}}) = \prod_{d_n \in \mathcal{N}} p(d_n). \tag{24}$$

We can develop the Bayesian model of stereo matching based on the independence assumption as above. Inserting *eq.* (24) into *eq.* (16),

$$p(d_{i,j}) = \sum_{\mathcal{N} \in \mathcal{S}} \left[p(d_{i,j} \mid \mathbf{d}_{\mathcal{N}}) \prod_{d_n \in \mathcal{N}} p(d_n) \right], \tag{25}$$

and, substitute $p(d_{i,j}|\mathbf{d}_{\mathcal{N}})$ in *eq.* (25) with *eq.* (15). By interchanging the summation and multiplication operator, the new and more practical general formula is derived in the probability space as follows.

$$p(d_{i,j}) = p_0(d_{i,j}) \sum_{\mathcal{N} \in \mathcal{S}} \left[\prod_{d_n \in \mathcal{N}} \exp\left\{-\rho_d(d_{i,j} - d_n)\right\} p(d_n) \right]. \tag{26}$$

Also, *eq* (26) can be transformed into the energy space with the Gibb's distribution as before.

$$E(d_{i,j}) = E_0(d_{i,j}) - \log \left[\sum_{\mathcal{N} \in \mathcal{S}} \left\{ \prod_{d_n \in \mathcal{N}} \exp\{-\rho_d(d_{i,j} - d_n)\} \, p(d_n) \right\} \right] \tag{27}$$

Eq (26) and (27) are the new general formula of Bayesian model for stereo matching based on the assumption of independence between the disparities in the configuration. This formula is much more useful for implementation than *eq.* (19), because the marginal probability $p(d_{i,j})$ can be calculated easily in the energy space. In addition, since the energy space is diffused iteratively with this formula, it is possible to estimate the marginal probability more accurately in each iteration step. Now, the only thing to do is to guarantee that the probability, which is transformed from energy space by Gibb's distribution, satisfy the usual probability property. That is to say, the probability space transformed from the ennergy space cannot satisfy the axiomatic properties of probability theory. This incomplete probability space can result in the divergence in the energy space as the diffusion processing is proceeded iteratively. So, it is necessary to set the constraint on the probability space as follows.

$$\sum_{d_{i,j} \in \mathcal{D}} p(d_{i,j}) = 1, \tag{28}$$

$$p(d_{i,j}) = \frac{\exp\{-E(d_{i,j})\}}{\sum_{d_{i,j} \in \mathcal{D}} \exp\{-E(d_{i,j})\}}. \tag{29}$$

As the previous manipulation based on the inequality between arithmetic mean and geometric mean, we manipulate the *eq.* (27) again as follows.

$$\begin{aligned}
& \log \left[\sum_{\mathcal{N} \in \mathcal{S}} \left\{ \prod_{d_n \in \mathcal{N}} \exp\{-\rho_d(d_{i,j} - d_n)\} \, p(d_n) \right\} \right] \qquad (30) \\
\geq \ & \log \left[n(\mathcal{S}) \sqrt[n(\mathcal{S})]{\prod_{\mathcal{N} \in \mathcal{S}} \left\{ \prod_{d_n \in \mathcal{N}} \exp\{-\rho_d(d_{i,j} - d_n)\} \, p(d_n) \right\}} \right] \\
= \ & \log n(\mathcal{S}) + \frac{1}{n(\mathcal{S})} \left[\sum_{\mathcal{N} \in \mathcal{S}} \left\{ \sum_{d_n \in \mathcal{N}} \log \{ \exp\{-\rho_d(d_{i,j} - d_n)\} \, p(d_n) \} \right\} \right] \\
= \ & \log n(\mathcal{S}) + \frac{1}{n(\mathcal{S})} \left[\sum_{\mathcal{N} \in \mathcal{S}} \left\{ \sum_{d_n \in \mathcal{N}} \{ \log(p(d_n)) - \rho_d(d_{i,j} - d_n) \} \right\} \right]
\end{aligned}$$

Finally, we can derive the new practical inequality based on the assumption of independence between the diparities of the configuration.

$$E(d_{i,j}) \leq E_0(d_{i,j}) - \log n(\mathcal{S}) \;-\; \frac{1}{n(\mathcal{S})}\left[\sum_{\mathcal{N}\in\mathcal{S}}\left\{\sum_{d_n\in\mathcal{N}}\{\,\log\left(p(d_n)\right) \;-\; \rho_d(d_{i,j}-d_n)\,\}\right\}\right] \tag{31}$$

As the same as *eq.* (23), we can see the disparity regularization and joint probability dependence. In this equation, it is different from the *eq.* (23) that marginal probability distribution have to be considered in order to diffuse the energy space, instead of joint probability distribution.

4 Proposed Algorithm

In this section, this paper proposes a stereo matching algorithm based on *eq.* (31) and assumption of similarity between disparities in the configuration. In *eq.* (31), we should choose the configuration and its elements $d_n{}'s$ so as to maximize the marginal probability $p(d_n)$. Since we have no any information of disparity map, however, we have no choice but to estimate the marginal probability. we assume that the disparity varies or distributes continuousely with higher probability than it does abruptly. As is the same case of the intensity distribution in the image, the disparity varies smoothly except for the boundaries of objects. Based on the above assumption, we can estimate the disparities in the configuration as the same as the disparity $d_{i,j}$. That is to say, if the disparity at (i,j) is $d_{i,j}$, the all the disparities in the configuration are estimated as the same as the $d_{i,j}$. This is reasonable estimation since we have no any information of the distribution of disparity, and disparities in the configuration are dependent on one another. In addition, this assumption is consistent with the disparity based regularization measure in the Bayesian model which makes the distribution of disparity smmoth and continuous. And, this paper uses the only one geometry of configuration. In this paper, the first geometry, Fig 2. (a) is used.

With the above assumption and geometry, we can change *eq.* (31) into new formmula. In this case, the number of possible configurations, $n(\mathcal{S})$ becomes equivalent to the number of disparity range, $n(\mathcal{D})$, because we choose all the disparities in the configuration as the same as the disparity $d_{i,j}$ at the position. So, the diffusion equation *eq.* (31) is changed into *eq.* (32),

$$E(d_{i,j}) \leq E_0(d_{i,j}) - \log n(\mathcal{D}) \;-\; \frac{1}{n(\mathcal{D})}\left[\sum_{\mathcal{N}\in\mathcal{S}}\left\{\sum_{d_n\in\mathcal{N}}\{\,\log\left(p(d_n)\right) \;-\; \rho_d(d_{i,j}-d_n)\,\}\right\}\right]. \tag{32}$$

And, the general formula is equivalent to *eq.* (27),

$$E(d_{i,j}) = E_0(d_{i,j}) - \log \left[\sum_{\mathcal{N} \in \mathcal{S}} \left\{ \prod_{d_n \in \mathcal{N}} \exp\left\{-\rho_d(d_{i,j} - d_n)\right\} p(d_n) \right\} \right] \tag{33}$$

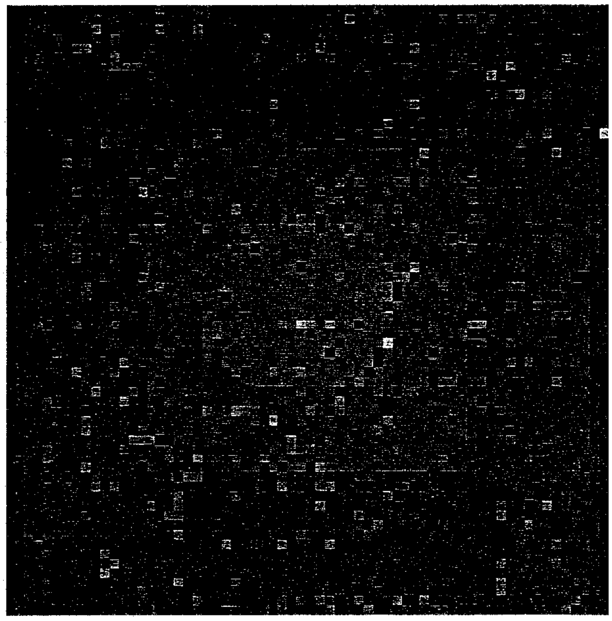
5 Experimental Results

This paper had various experiments with *eq.* (33) and 3 stereo images sets, which consist of 5 images generated from 5 different viewpoints respectively. The reference image is always the center image in each stereo images set. The five images have been rectified and satisfy the epipolar geometry. Before comparing the performance of the proposed algorithm with that of the conventional one, we can understand how the formula of Bayesian model for stereo matching operate on the initial energy space iteratively in Fig. 3. The first figure is the initial energy space and next figures are the first, second, third, 4th, 6th, 9th, and 15th disparity map, respectively. As we can see in the successive figures, the disparity is converged to the neighborhood disparity with high probabilty. In this way, the disparity map is decided in some iterations.

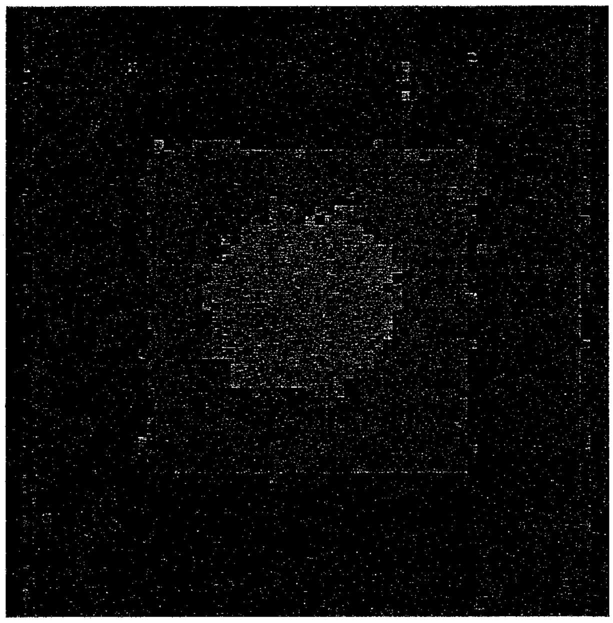
Fig. 4 is the center image of the *randomdot* images set and Fig. 5 is the true disparity map. So is the *face* image in Fig. 6 and 7. And, Fig. 8 is the center image of *doll* whose disparity map is not known. With these stereo images sets, two kinds of experimental results are compared. The first experiment compares the performance of the conventional algorithm [?, ?] with that of the proposed one. The second experiment compares the estimated disparity map from two viewpoints images set, which is the usual case, with the estimated disparity map from five viewpoints images set.

5.1 Experiments of the Proposed Algorithm

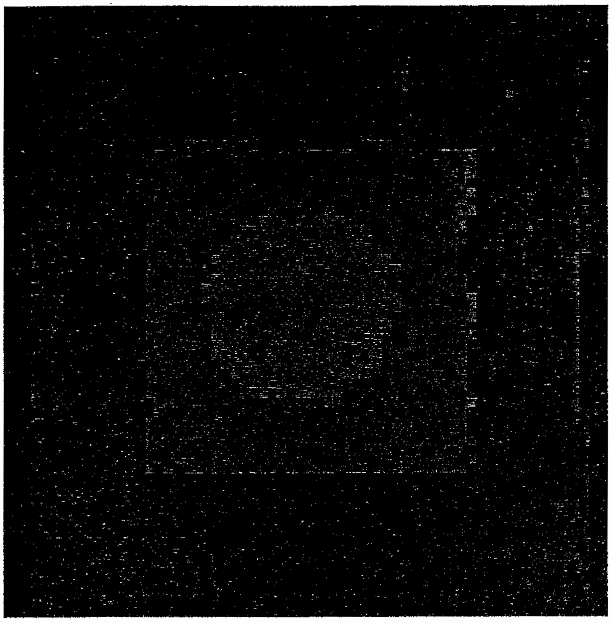
In this section, the paper summarizes the experimental results. In order to evaluate the performance, the paper compares the disparity maps generated from the proposed algorithm with those of the SSD and conventional algorithm[?, ?]. The conventional algorithm is also based on the independence of marginal distribution. However, the algorithm is confined to the specific probability model such as the unique configuration and ambiguous independence assumption. In other words, the algorithm used only one configuration of Fig. 2 (a), and assumed the independence not between marginal distributions, but between averaged distributions in the disparity range.



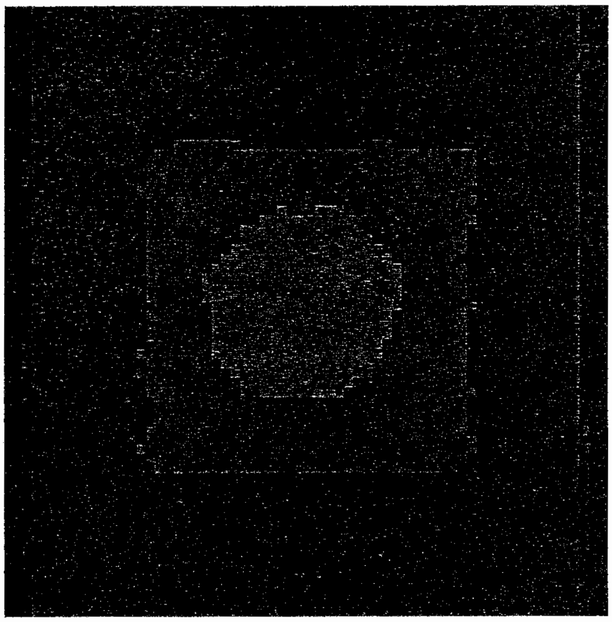
(a) initial disparity map



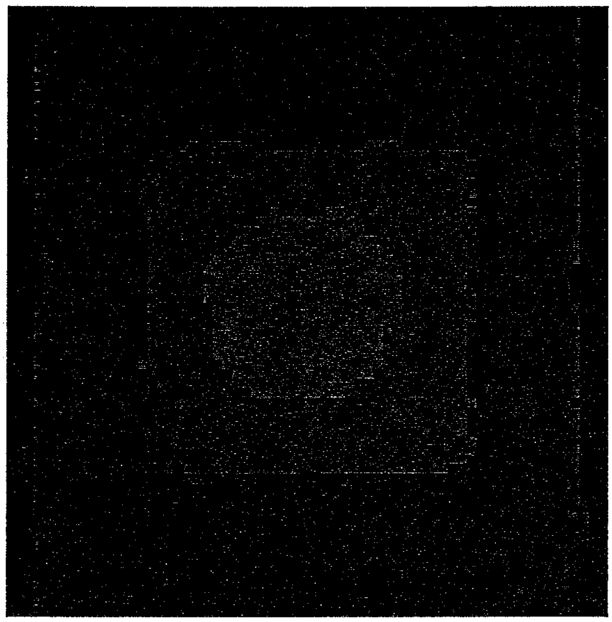
(b) once iterated disparity map



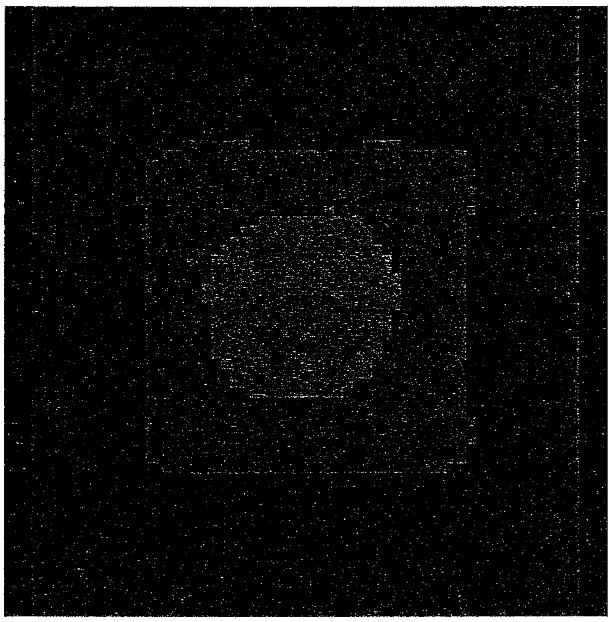
(c) twice iterated disparity map



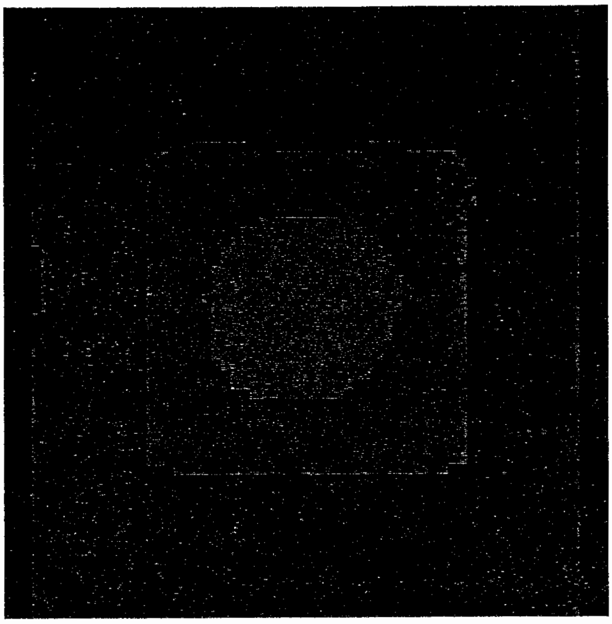
(d) three times iterated disparity map



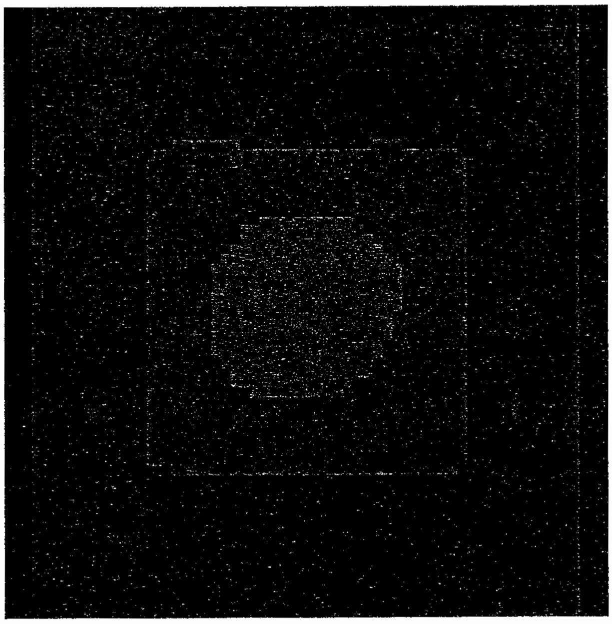
(e) four times iterated disparity map



(f) five times iterated disparity map



(g) nine times iterated disparity map



(h) fifteen times iterated disparity map

Figure 3: Convergence of the disparity map

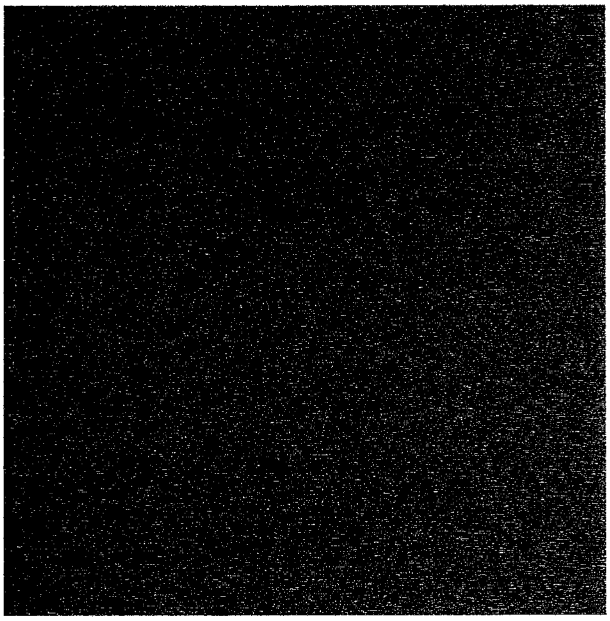


Figure 4: Test stereo image *random dot*

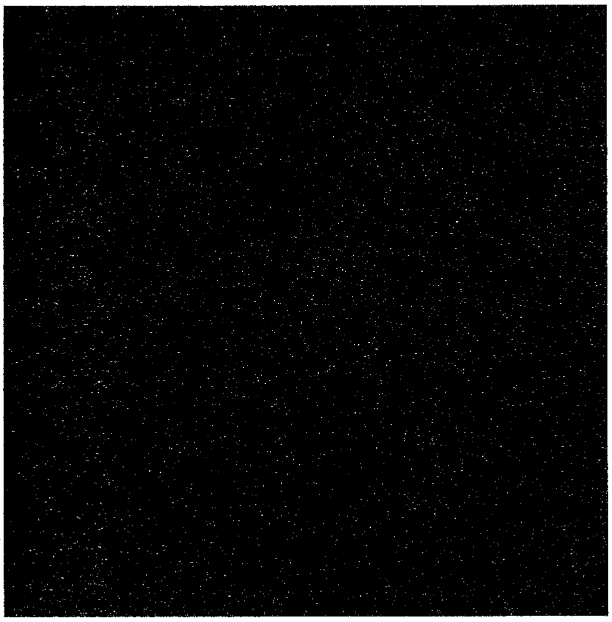


Figure 5: True disparity map of *random dot*

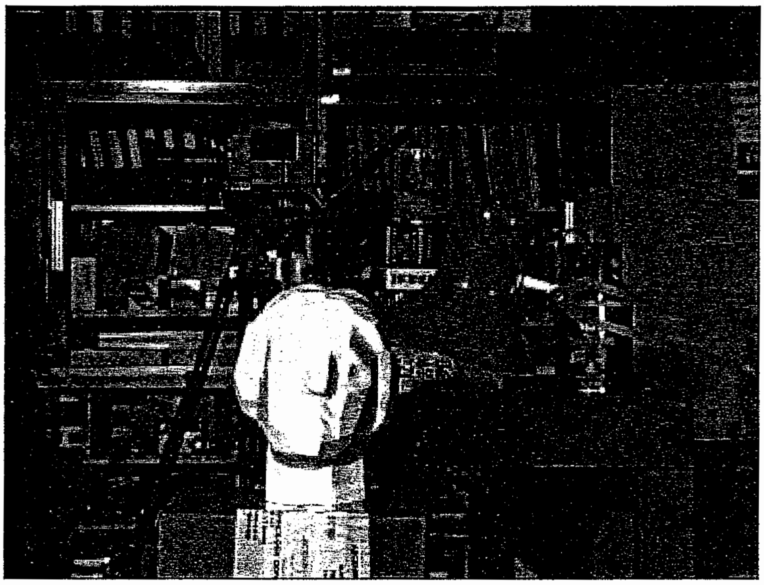


Figure 6: Test stereo image *face*

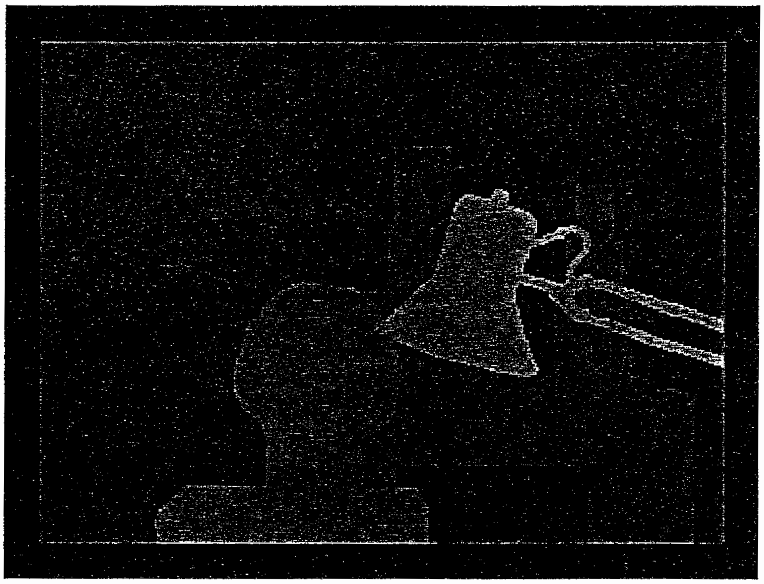


Figure 7: True disparity map of *face*

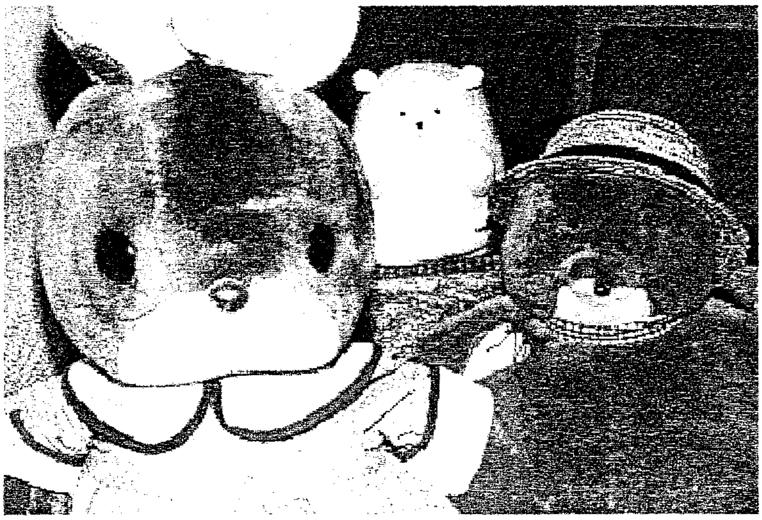


Figure 8: Test stereo image *doll*

$$E(d_{i,j}) = E_0(d_{i,j}) - \sum_{d_n \in \mathcal{N}} \log \left[\sum_{\mathcal{N} \in \mathcal{S}} \exp\{-\rho_d(d_{i,j} - d_n)\} p(d_n) \right] \quad (34)$$

The *eq.* (34) can be derived from the assumption as following conditions.

$$p(x_1, x_2) = \left[\sum_{x_1} p(x_1) \right] \left[\sum_{x_2} p(x_2) \right] \quad (35)$$

This paper has implemented the proposed algorithm in *eq.* (33) with the same configuration and all the same parameters in order to compare the performances and probability model with those of *eq.* (34). In the experiments, the energy space was generated from 5 viewpoints stereo images. The four energy spaces generated from center image and the other four images are averaged in each three dimensional energy space. Fig. 9, 10, and 11 show the comparison of two disparity maps for *random dot* generated from three algorithms respectively. The Fig. 9 is generated from the SSD of block size 7x7, Fig. 10 is generated from the conventional algorithm and the Fig. 11 from the proposed one. As we can see, the disparity map from the proposed algorithm is superior to those from the SSD and conventional one. The disparity boundares of the Fig. 11 are closer to the true map, and the false disparities are fewer than the Fig. 9 and 10. In the three figure, the left and right areas result from the modulo operation in the implemetation program. So, they are meaningless regions in the disparity maps. Fig. 12, 13, and 14 show the disparity maps for *face*

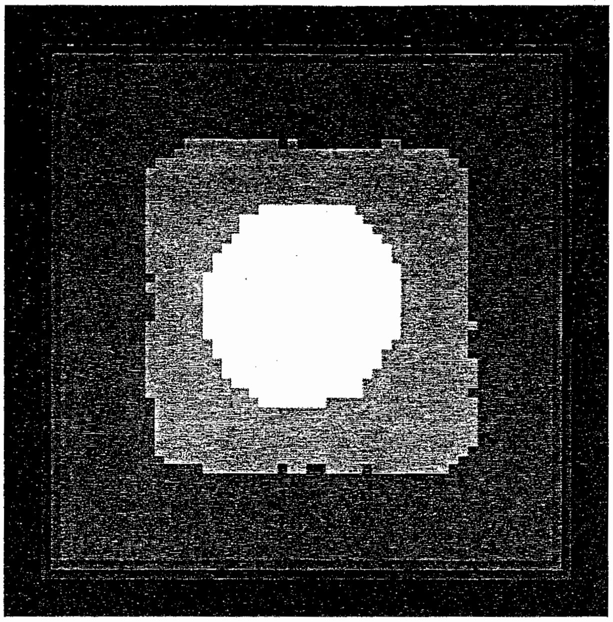


Figure 9: Disparity map from SSD

images, and Fig. 15 , 16, and 17 for *doll* images. As are the same results of the *random dot* images, the proposed algorithm outperforms the conventional one, in that the clearer and more accurate boundaries, and the fewer errorneous disparities. In any stereo images, the proposed algorithm generates the better disparity map than the conventional algorithm. This result shows that the assumption of probabilty and similarity of disparity, which was described in section 4., is reasonable and sufficeint for the estimation of the joint probability distribution of configuration.

5.2 Experiments of Multiple Viewpoints Stereo Images

In this section, the paper shows the various experimental results with five viewpoints stereo images. As is different from the usual case of two stereo images, the stereo images generated from five different viewpoints are used. In the first place, this paper shows the improvement of the performance using multiple stereo images instead of two stereo images. And then, this paper will compare with the disparity maps from different generation methods of energy space. Fig 18 shows the disparity map from only two stereo images, center and left images of *random dot*. Fig. 19 is the disparity map from three images, center, left, and right images. Compared the two disparity maps with Fig. 11, which is generated from five stereo images , as we can expect,

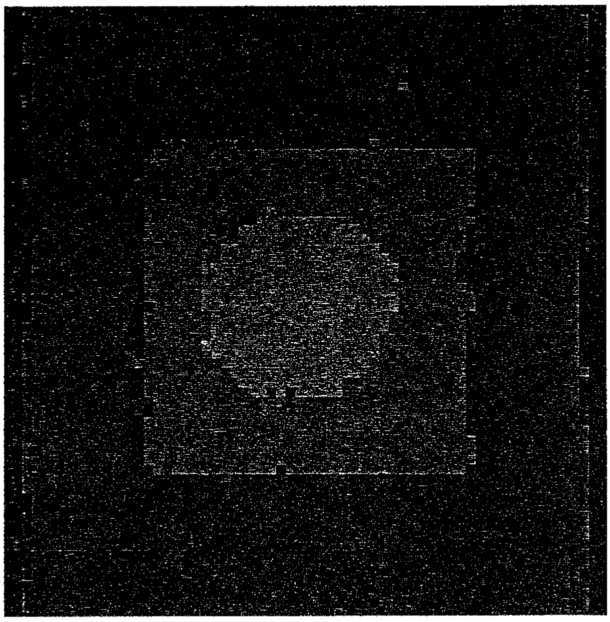


Figure 10: Disparity map from Conventional[3,4]

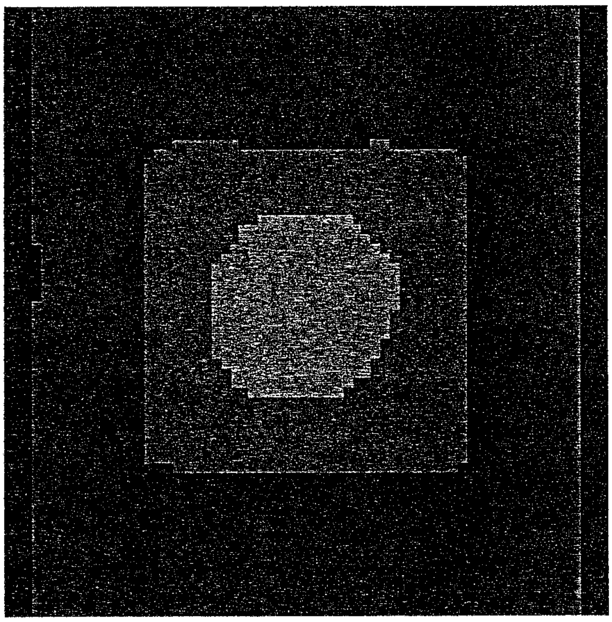


Figure 11: Disparity map from Proposed

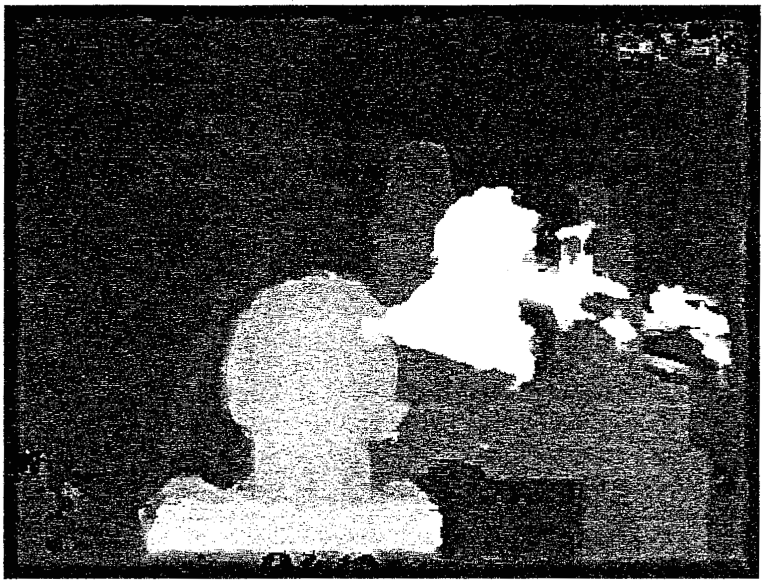


Figure 12: Disparity map from SSD

Figure 13: Disparity map from Conventional[3,4]

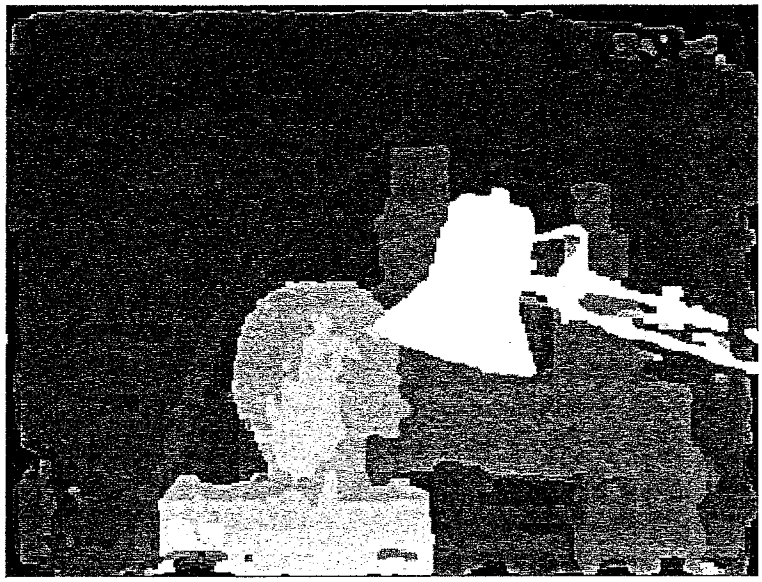


Figure 14: Disparity map from Proposed

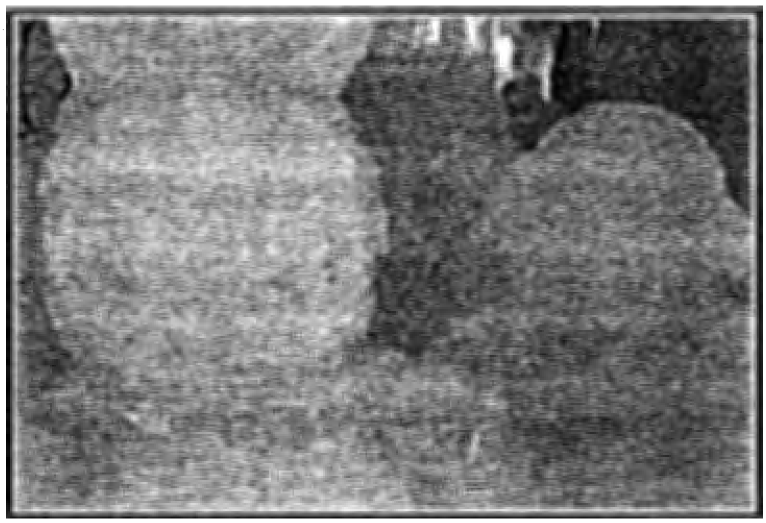


Figure 15: Disparity map from SSD

Figure 16: Disparity map from Conventional[3,4]

Figure 17: Disparity map from Proposed

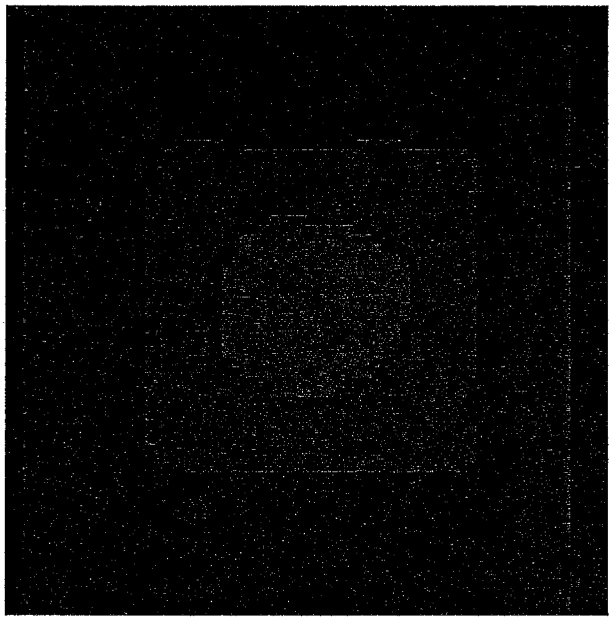


Figure 18: Disparity map from two stereo images, center, left

the more stereo images, the better disparity map is generated. In addition, the iteration number is reduced in the case of multiple stereo images. In other words, the convergence becomes faster as increasing the stereo images. This is because of the better initial energy space generated from multiple stereo images. However, multiple images need many memory space and complex rectification among the stereo images. In Fig. 20 and 21 of $face$, we can show the same improvement by comparing two figures with Fig. 14. Also, comparing Fig. 22 and 23 with Fig. 17, we can see that the more stereo images generate the better disparity map.

Now, the next experiments compare with the methods to generate the initail energy space. In these experiments, the main problem is how to generate the initial energy space from five stereo images. This paper had experiments with three methods. The first experiment makes the energy space by averaging the four energy spaces from the center and the other four images. This method is also applied to the comparison experiments in section 5.1. This method is called *averaging method.* The second makes the energy space by selecting the minimum energy among the four energy spaces at each position. This is called *minimum method.* And the final method makes the energy space by selecting one minimum energy space in the both horizontal and vertical direction respectively, and averaging them. This is called

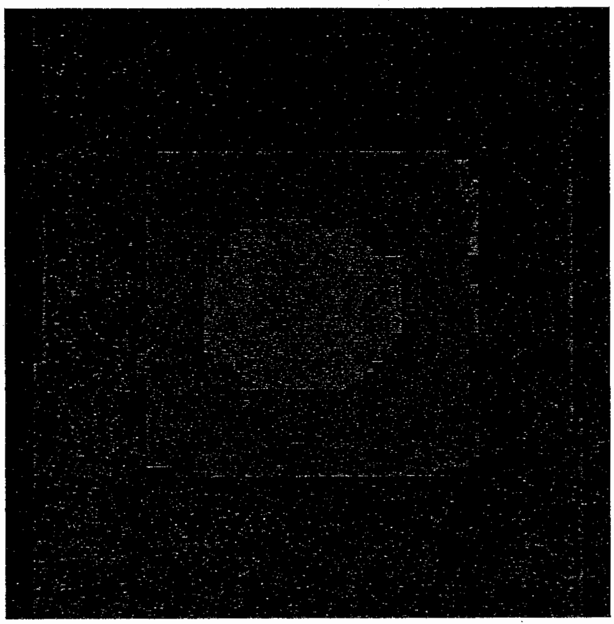


Figure 19: Disparity map from three stereo images, center, left, right

Figure 20: Disparity map from two stereo images, center, left

Figure 21: Disparity map from three stereo images, center, left, right

directional minimum method. The experimental results of disparity map are compared with one another from Fig. 24 to 29. The Fig. 24 and 25 are the results of *minimum method* and *directional minimum method* for *random dot*. The Fig. 26 and 27 are the results for the *face* and Fig. 30 abd 31 for *doll*. Compared with the Fig. 11, 14 and 17 for each disparity map from *averaging method*, respectively, the other two methods have poor performances and generate almost same disparity maps. In order to compare with the initial energy spaces easily ,we can see the difference of initializations in Fig. 28 and 29 for *face* and Fi. 32 and 33 for *doll*. This result show that it is important to generate the initial energy space as good as possible, and noisy generation such as *minimum method* and *directional minimum method* are not adequate for the initial energy space generation.

6 Conclusions

This paper has derived the general formula of Bayesian model for stereo matching. It has implemented the proposed algorithm based on the independence of probability between disparities in the configuration and similarity of disparity assumption. Also, this paper had various experiments with multiple viewpoints stereo images, which is differnt from the usual case of two stereo images. According to the experimental

Figure 22: Disparity map from two stereo images, center, left

results, the performance of the proposed algorithm outperformed the conventional algorithm [?, ?], in that the boundaries of disparity were clearer and errorneous disparity was fewer than the conventional algorithm. This result means that the derived formula is adequate and the assumptions to derive the formula are reasonable and sufficient for the joint probability distribution. In the experiments of various initialization of energy spaces, it was sure that multiple viewpoints stereo images made the more accurate disparity map than two or fewer stereo images did. This result means that the multiple stereo images will be very useful to estimate the more acuurate disparity map in stereo matching and to solve the occlusion problems in view synthesis. Moreover, in Bayesian model of stereo matching, it is very important to generate the better initial energy space given stereo images set.

Finally, we can conclude the following facts according to the experimental results. The first is that the derived formula is the general form and can be changed into the some different forms based on the reasonable probabilistic assumptions. The more accurate is the assumed probability model, the better disparity map can be generated with this formula. And, this Bayesian model can be developed with various probabilistic model and configuration. The second is that it is very important to generate the initial energy space in Bayesian model of stereo matching, so the multiple stereo images are useful for the estimation of the better disparity map. In order to improve this formula, the research for estimation of exact joint distribution and

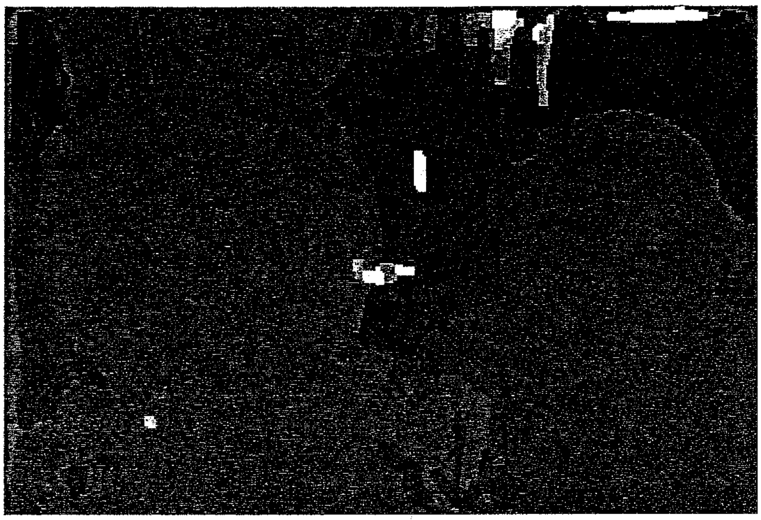


Figure 23: Disparity map from three stereo images, center, left, right

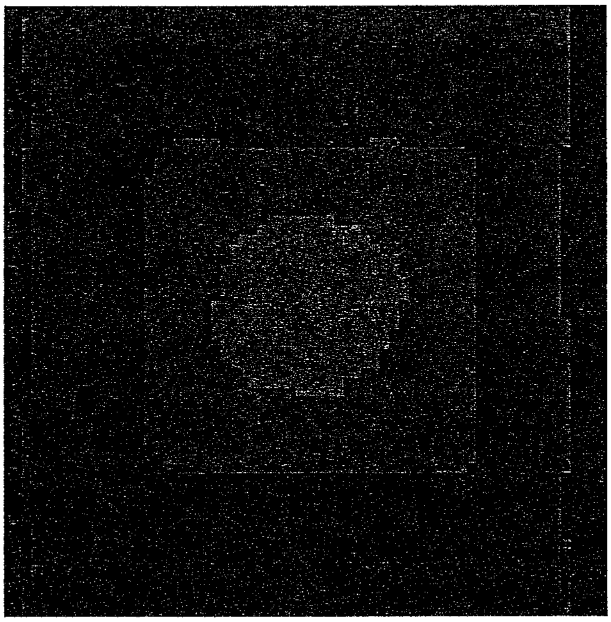


Figure 24: Disparity map from minimum energy method

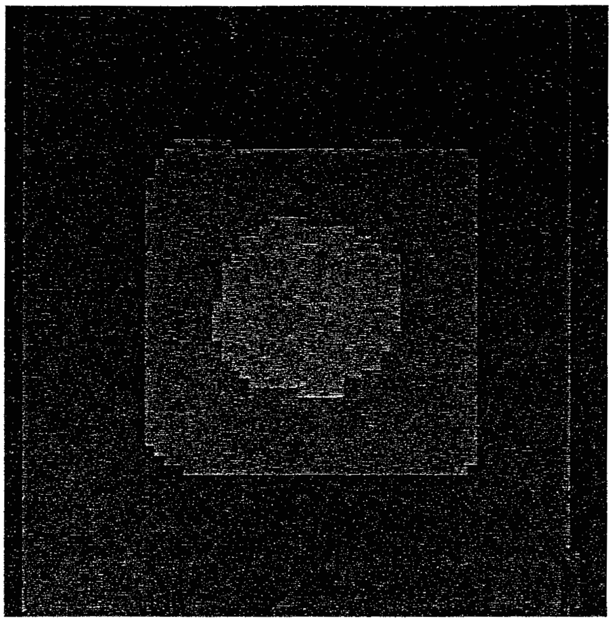


Figure 25: Disparity map from directional minimum energy method

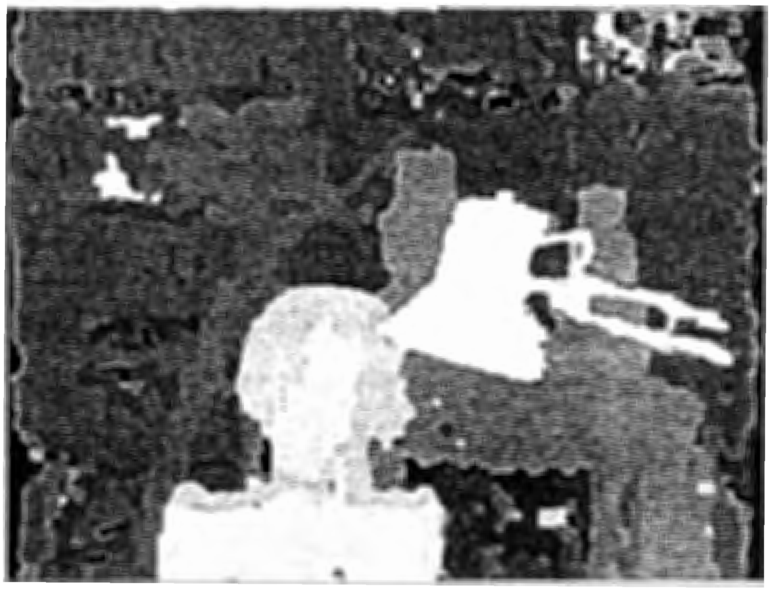


Figure 26: Disparity map from minimum energy method

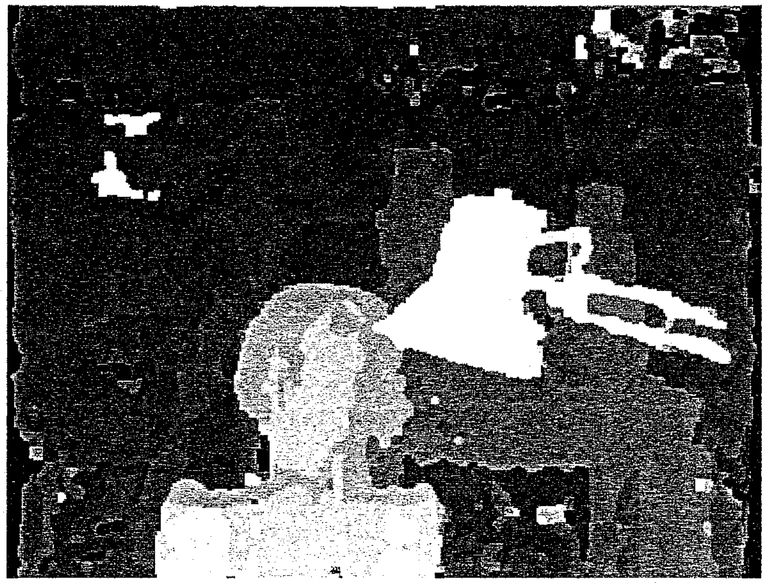


Figure 27: Disparity map from directional minimum energy method

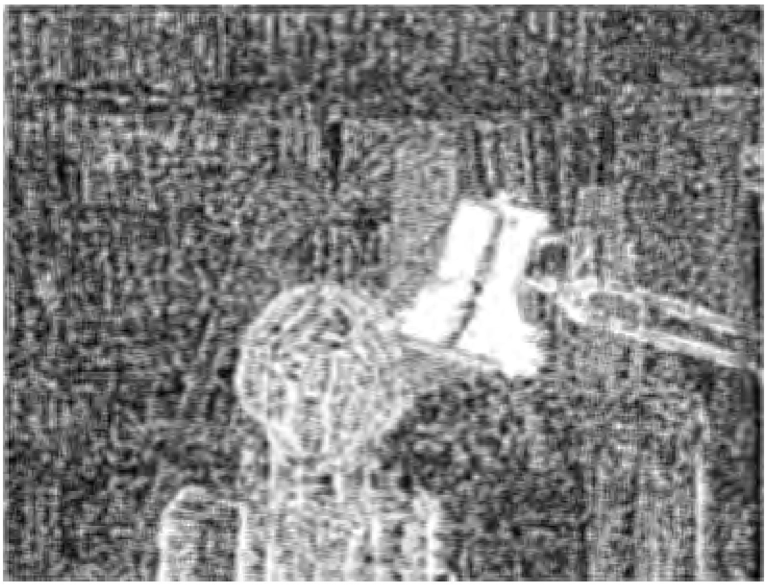


Figure 28: Disparity map from initial energy space by *minimum method*

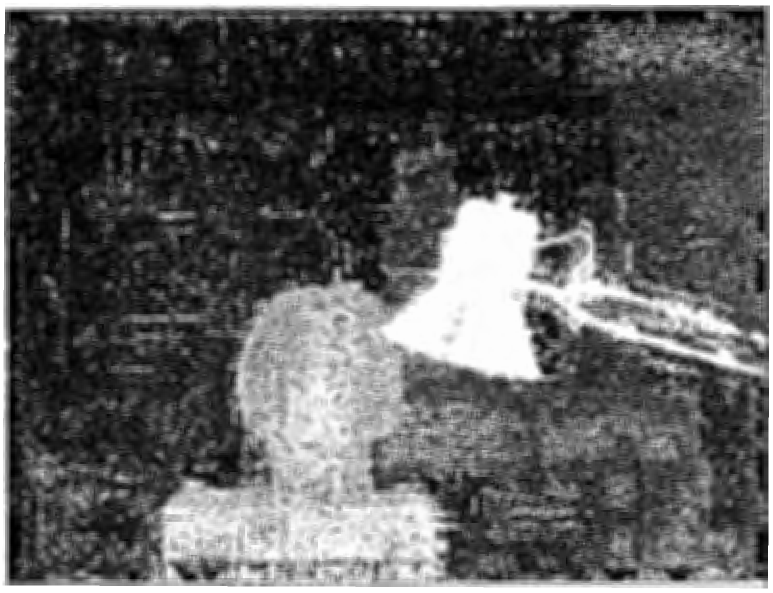


Figure 29: Disparity map from initial energy space by *averaging*

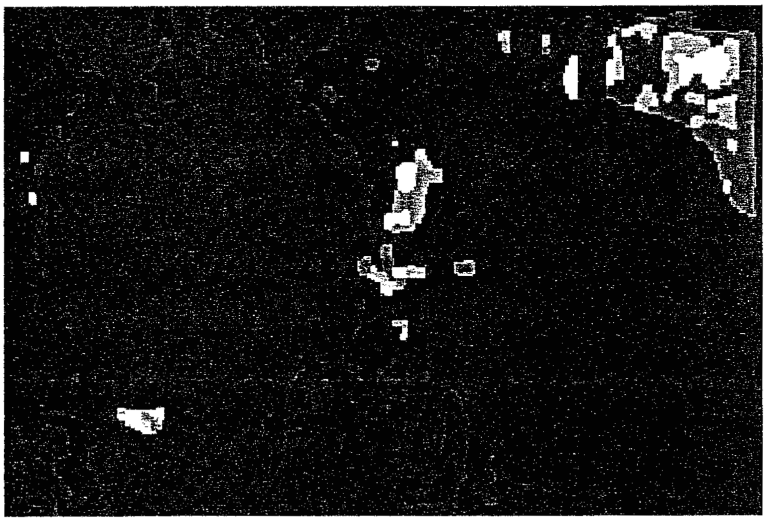


Figure 30: Disparity map from minimum energy method

Figure 31: Disparity map from directional minimum energy method

Figure 32: Disparity map from initial energy space by *minimum method*

Figure 33: Disparity map from initial energy space by *averaging*

various configurations will be necessary. Also, the analysis of the convergence should be described to complete the formulation.

7 Acknowledgement

I came to ATR when the mountain and plain were green, and I am going to Korea when they are golden. As much as the nature has grown for the last two months, I also have grown for the period and learned many things. I won't forget the life in Japan and with my colleagues in Department 3, MI&C for all my life. The two months was very short period to do many things and to get along friendly with my colleagues in Department 3, MI&C. In this acknowledgement, I'd like to appreciate all of them to help me and to get along with me in Japan for the two months. And, most of all, I appreciate my senior, Jong-Il Park, who gave me a chance to come here and helped me in everything. Without him, I would never have had this beautiful and good experience. Thank him sincerely again.

I wish my senior and all my colleagues in Department 3, MI&C be happy and healthy. Also, I hope to meet the colleagues with pleasure sometimes.

References

[1] Rafael C. Gonzalez, *Digital Image Processing.* Addison Wesley, 1994.

[2] Erwin Kreyzig, *Introductory Functional Analysis with Applications* . WIE Wiley, 1978.

[3] D. Scharstein and R. Szeliski, "Stereo Matching with Non-Linear Diffusion," *Proc. of In IEEE Computer Society Conference on Computer Visio and Pattern Recognition (CVPR-96)* , pp. 343-350, San Francisco, CA, June 1996.

[4] D. Scharstein, *View Synthesis Using Stereo Vision* , Dissertation for Ph. D Degree, Cornell University, Feb. 1997.

[5] S. Geman and D. Geman, "Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images," *IEEE Trans. on Pattern Analysis and Machine Intelligence* , Vol. PAMI-6, No. 6, pp. 721-741, Nov. 1984.

[6] D. Scharstein, "Matching Images by Comparing their gradient fields," *Proc. of International Conference on Pattern Recognition(ICPR'94)* Vol. 1, pp. 572-575, Oct. 1994.

```
/*************************************************************/
/*                                                           */
/*    Stereo Matching with Diffusion Program                 */
/*    Sang Hwa  Lee,     1997.  8.  7.  in ATR               */
/*                                                           */
/*    Multiview(5 Camera Images)                             */
/*    Disparity Space from Weighted 4 Images                 */
/*    Point based initial disparity space construction       */
/*    Stopping Condition Based on Disparity Energy Variation */
/*    New Derived Update Equation (Pure Version)             */
/*    Momentum Addition: Convergence Improvement             */
/*                                                           */
/*************************************************************/

#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include "memory.h"
#include "in_out.h"

#define  VAR_P  0.32     /* Variance of prior contaminated Gaussian   */
#define  EPS_P  0.01     /* Epsilon of prior contaminated Gaussian    */
#define  VAR_M  50.0     /* Variance of measure contaminated Gaussian */
#define  EPS_M  0.1      /* Epsilon of measure contaminated Gaussian  */
#define  D_max  15       /* Maximum disparity                         */
#define  D_min  0        /* Minimum disparity                         */
#define  WS     1
#define  STOP   0.0004  /* Stopping Condition of Iteration           */

float        wr, wl, wt, wb;


float P_Contaminated_Gauss ( int x )
{
  return (  -flog( (1.0-EPS_P)*fexp((-x*x)/(VAR_P)) + EPS_P )  );
}



float M_Contaminated_Gauss ( int x )
{
  return (  -flog( (1.0-EPS_M)*fexp((-x*x)/(VAR_M)) + EPS_M )  );
}



Find_Local_Space ( float ***ds, float *local, int i, int j )
{
  int          k;

  for ( k=0 ; k <= D_max+D_min ; k++ )
    local[k] = ds[k][i][j];
}



int Minimum_Disparity ( float *local )
{
  int          k, index_min;
  float        minimum;

  minimum = local[0];
  index_min = 0;

  for ( k=1 ; k <= D_max+D_min ; k++ )
    if ( minimum > local[k] )
```

```
/****************************************************************/
/*                                                              */
/*     Stereo Matching with Diffusion Program                   */
/*     Sang Hwa  Lee,     1997.  8.  7.  in ATR                 */
/*                                                              */
/*     Multiview(5 Camera Images)                               */
/*     Disparity Space from Weighted 4 Images                   */
/*     Point based initial disparity space construction         */
/*     Stopping Condition Based on Disparity Energy Variation   */
/*     New Derived Update Equation (Pure Version)               */
/*     Momentum Addition: Convergence Improvement               */
/*                                                              */
/****************************************************************/

#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include "memory.h"
#include "in_out.h"

#define  VAR_P  0.32     /* Variance of prior contaminated Gaussian   */
#define  EPS_P  0.01     /* Epsilon of prior contaminated Gaussian    */
#define  VAR_M  50.0     /* Variance of measure contaminated Gaussian */
#define  EPS_M  0.1      /* Epsilon of measure contaminated Gaussian  */
#define  D_max  15       /* Maximum disparity                         */
#define  D_min  0        /* Minimum disparity                         */
#define  WS     1
#define  STOP   0.0004  /* Stopping Condition of Iteration          */

float        wr, wl, wt, wb;


float P_Contaminated_Gauss ( int x )
{
  return (  -flog( (1.0-EPS_P)*fexp((-x*x)/(VAR_P)) + EPS_P )  );
}



float M_Contaminated_Gauss ( int x )
{
  return (  -flog( (1.0-EPS_M)*fexp((-x*x)/(VAR_M)) + EPS_M )  );
}



Find_Local_Space ( float ***ds, float *local, int i, int j )
{
  int           k;

  for ( k=0 ; k <= D_max+D_min ; k++ )
    local[k] = ds[k][i][j];
}



int Minimum_Disparity ( float *local )
{
  int           k, index_min;
  float         minimum;

  minimum = local[0];
  index_min = 0;

  for ( k=1 ; k <= D_max+D_min ; k++ )
    if ( minimum > local[k] )
```

```
    {
      index_min = k;
      minimum = local[k];
    }

  return (index_min);
}



float Calculate_Sum_Prob ( float *local )
{
  int          k;
  float        temp=0.0;

  for ( k=0 ; k <= D_max+D_min ; k++ )
    temp += fexp( -local[k] );

  return ( temp );
}


float Diffusion (float ***ds, float ***prob, int d, int i, int j, int xsz, int ysz)
{
  int          k, x_off, y_off, num;
  float        diffuse, neighbor;

  diffuse = 0.0;
  for ( k=0 ; k <= D_max+D_min ; k++ )
  {
    neighbor = 1.0;
    for ( num=0 ; num < 4 ; num++ )
    {
      switch ( num )
      {
        case  0 :
                x_off = 1;
                y_off = 0;
                break;
        case  1 :
                x_off = -1;
                y_off = 0;
                break;
        case  2 :
                x_off = 0;
                y_off = 1;
                break;
        case  3 :
                x_off = 0;
                y_off = -1;
                break;
        default :  ;
      }
      neighbor *= fexp( -P_Contaminated_Gauss(d-k) )
                           * prob[k][(i+y_off+ysz)%ysz][(j+x_off+xsz)%xsz];
    }
    diffuse += neighbor;
  }
  return ( -flog(diffuse) );
}



float ENTROPY ( float *p )
{
  int          d;
```

```
  float         sum, entropy;

  sum = entropy = 0.0;

  for ( d=0 ; d <= D_min+D_max ; d++ )
  {
    p[d] = fexp(-p[d]);
    sum += p[d];
  }

  for ( d=0 ; d <= D_min+D_max ; d++ )
    entropy += -(p[d]/sum)*flog((p[d]/sum));

  return ( entropy );
}



Block_Init_Disparity_Space (float ***ds, unsigned char ***img, int xsz, int ysz)
{
  int           d, i, j, k, l, img_num;
  float         sum[4], temp, diff;

  for ( d=-D_min ; d <= D_max ; d++ )
    for ( i=0 ; i < ysz ; i++ )
      for ( j=0 ; j < xsz ; j++ )
      {
        for ( img_num=1 ; img_num <= 4 ; img_num++ )
        {
          temp = 0.0;

          if ( img_num == 1 && wr != 0.0 )
            for ( k=-WS ; k <= WS ; k++ )
              for ( l=-WS ; l <= WS ; l++ )
              {
                diff = img[0][(ysz+i+k)%ysz][(xsz+j+l)%xsz]
                       - img[img_num][(ysz+i+k)%ysz][(xsz+j+l-d)%xsz];
                temp += M_Contaminated_Gauss (diff);
              }

          else if ( img_num == 2 && wl != 0.0 )
            for ( k=-WS ; k <= WS ; k++ )
              for ( l=-WS ; l <= WS ; l++ )
              {
                diff = img[0][(ysz+i+k)%ysz][(xsz+j+l)%xsz]
                       - img[img_num][(ysz+i+k)%ysz][(xsz+j+l+d)%xsz];
                temp += M_Contaminated_Gauss (diff);
              }

          else if ( img_num == 3 && wt != 0.0 )
            for ( k=-WS ; k <= WS ; k++ )
              for ( l=-WS ; l <= WS ; l++ )
              {
                diff = img[0][(ysz+i+k)%ysz][(xsz+j+l)%xsz]
                       - img[img_num][(ysz+i+k+d)%ysz][(xsz+j+l)%xsz];
                temp += M_Contaminated_Gauss (diff);
              }

          else if ( img_num == 4 && wb != 0.0 )
            for ( k=-WS ; k <= WS ; k++ )
              for ( l=-WS ; l <= WS ; l++ )
              {
                diff = img[0][(ysz+i+k)%ysz][(xsz+j+l)%xsz]
                       - img[img_num][(ysz+i+k-d)%ysz][(xsz+j+l)%xsz];
                temp += M_Contaminated_Gauss (diff);
              }
```

```
          else
            ;

          sum[img_num-1] = temp;
        }
        ds[D_min+d][i][j]=(wr*sum[0]+wl*sum[1]+wt*sum[2]+wb*sum[3])/(wr+wl+wt+wb);
      }
}



Point_Init_Disparity_Space (float ***ds, unsigned char ***img, int xsz, int ysz)
{
  int          d, i, j, img_num;
  float        sum[4], diff;

  for ( d=-D_min ; d <= D_max ; d++ )
    for ( i=0 ; i < ysz ; i++ )
      for ( j=0 ; j < xsz ; j++ )
      {
        for ( img_num=1 ; img_num <= 4 ; img_num++ )
        {
          if ( img_num == 1 && wr != 0.0 )
          {
            diff = img[0][(ysz+i)%ysz][(xsz+j)%xsz]
                 - img[img_num][(ysz+i)%ysz][(xsz+j-d)%xsz];
            diff = M_Contaminated_Gauss (diff);
          }

          else if ( img_num == 2 && wl != 0.0 )
          {
            diff = img[0][(ysz+i)%ysz][(xsz+j)%xsz]
                 - img[img_num][(ysz+i)%ysz][(xsz+j+d)%xsz];
            diff = M_Contaminated_Gauss (diff);
          }

          else if ( img_num == 3 && wt != 0.0 )
          {
            diff = img[0][(ysz+i)%ysz][(xsz+j)%xsz]
                 - img[img_num][(ysz+i+d)%ysz][(xsz+j)%xsz];
            diff = M_Contaminated_Gauss (diff);
          }

          else if ( img_num == 4 && wb != 0.0 )
          {
            diff = img[0][(ysz+i)%ysz][(xsz+j)%xsz]
                 - img[img_num][(ysz+i-d)%ysz][(xsz+j)%xsz];
            diff = M_Contaminated_Gauss (diff);
          }

          else
            ;

          sum[img_num-1] = diff;
        }
        ds[D_min+d][i][j]=(wr*sum[0]+wl*sum[1]+wt*sum[2]+wb*sum[3])/(wr+wl+wt+wb);
      }
}


int Stop_Condition (unsigned char **pre, unsigned char **cur, int xsz, int ysz)
{
  int            i, j, condition;
  float          energy, diff, pre_diff;
```

```
  diff = energy = 0.0;
  condition = 1;
  pre_diff = 1.0;

  for ( i=0 ; i < ysz ; i++ )
    for ( j=0 ; j < xsz ; j++ )
    {
      diff += (float)((pre[i][j]-cur[i][j])*(pre[i][j]-cur[i][j]))/100.;
      energy += (float)(pre[i][j]*pre[i][j])/100.;
    }

  if ( diff/energy < STOP || pre_diff < (diff/energy) - pre_diff*0.1 )
    condition = 0;

  pre_diff = diff/energy;
  printf("%f\n", diff/energy);
  return ( condition );
}



main ( int argc, char *argv[] )
{
  int             d, i, j, k, l, m, condition;
  float           ***Eo, ***E, ***tmp_E, ***prob, ***moment, *tmpbuf, tmp, Diff;
  unsigned char   ***in, **out, **dp, **dp_pre;
  int             xsz, ysz, iter;
  FILE            *fp;
  char            output[30];

  if ( argc != 14 )
  {
    printf("USAGE: Command 5_input_images xsz ysz iter\n       ");
    printf("Order of 5_input_images: center, right, left, top, bottom\n");
    printf("       output, x_size, y_size, iteration_number\n");
    exit (1);
  }

  xsz  = atoi ( argv[7] );
  ysz  = atoi ( argv[8] );
  iter = atoi ( argv[9] );
  wr   = atof ( argv[10] );
  wl   = atof ( argv[11] );
  wt   = atof ( argv[12] );
  wb   = atof ( argv[13] );

  in = Memory_3D_unsigned_char ( xsz, ysz, 5 );
  Eo = Memory_3D_float ( xsz, ysz, D_max+D_min+1 );

  for ( k=0 ; k < 5 ; k++ )
    fread_2D( in[k], xsz, ysz, argv[k+1] );

  Point_Init_Disparity_Space ( Eo, in, xsz, ysz );
  free ( in );

  E      = Memory_3D_float ( xsz, ysz, D_max+D_min+1 );
  tmp_E  = Memory_3D_float ( xsz, ysz, D_max+D_min+1 );
  prob   = Memory_3D_float ( xsz, ysz,  D_max+D_min+1 );
  moment = Memory_3D_float ( xsz, ysz, D_max+D_min+1 );
  tmpbuf = (float *)calloc( D_max+D_min+1, sizeof(float) );
  dp     = Memory_2D_unsigned_char ( xsz, ysz );
  dp_pre = Memory_2D_unsigned_char ( xsz, ysz );

  for ( d=0 ; d <= D_max+D_min ; d++ )
    for ( i=0 ; i < ysz ; i++ )
      for ( j=0 ; j < xsz ; j++ )
```

```
      tmp_E[d][i][j] = E[d][i][j] = Eo[d][i][j];

  for ( i=0 ; i < ysz ; i++ )
    for ( j=0 ; j < xsz ; j++ )
    {
      Find_Local_Space ( E, tmpbuf, i, j );
      dp[i][j]=(unsigned char)((250/(D_max+D_min))*Minimum_Disparity(tmpbuf));
    }

  fp = fopen ( "__name", "w" );
  fprintf(fp, "%s_init", argv[6]);
  fclose ( fp );
  fp = fopen ( "__name", "r" );
  fgets( output, 29, fp );
  fclose ( fp );

  fp = fopen ( output, "wb" );
  for ( i=0 ; i < ysz ; i++ )
    fwrite ( dp[i], sizeof(unsigned char), xsz, fp );
  fclose ( fp );

  m = 1;
  condition = 1;

  do
  {
    printf("%d/%d iteration is prosessing\n",m, iter);

    for ( i=0 ; i < ysz ; i++ )
      for ( j=0 ; j < xsz ; j++ )
      {
        Find_Local_Space ( E, tmpbuf, i, j );
        tmp = Calculate_Sum_Prob ( tmpbuf );
        for ( d=0 ; d <= D_max+D_min ; ++d )
          prob[d][i][j] = exp(-E[d][i][j]) / tmp;
      }

    for ( d=0 ; d <= D_min+D_max ; d++ )
      for ( i=0 ; i < ysz ; i++ )
        for ( j=0 ; j < xsz ; j++ )
        {
          Diff = Diffusion ( E, prob, d, i, j, xsz, ysz );
          tmp_E[d][i][j] = Eo[d][i][j]+Diff+0.2*moment[d][i][j];
          moment[d][i][j] = Diff;
        }

    for ( d=0 ; d <= D_min+D_max ; d++ )
      for ( i=0 ; i < ysz ; i++ )
        for ( j=0 ; j < xsz ; j++ )
          E[d][i][j] = tmp_E[d][i][j];


    for ( i=0 ; i < ysz ; i++ )
      for ( j=0 ; j < xsz ; j++ )
      {
        dp_pre[i][j] = dp[i][j];
        Find_Local_Space ( E, tmpbuf, i, j );
        dp[i][j]=(unsigned char)((250/(D_max+D_min))*Minimum_Disparity(tmpbuf));
      }

    if ( (m%2) == 0 && m >= 15 )
      condition = Stop_Condition ( dp_pre, dp, xsz, ysz );
    else
      condition = 1;

    if ( condition == 0 || m >= iter )
```

```
    {
      fp = fopen ( "__name", "w" );
      fprintf(fp, "%s_%d", argv[6], m);
      fclose ( fp );
      fp = fopen ( "__name", "r" );
      fgets( output, 29, fp );
      fclose ( fp );
      fp = fopen ( output, "wb" );

      for ( i=0 ; i < ysz ; i++ )
        fwrite ( dp[i], sizeof(unsigned char), xsz, fp );
      fclose ( fp );
    }
    m++;
  }
  while ( m <= iter && condition == 1 );

  free ( tmp_E );
  free ( prob );
  free ( Eo );
  free ( E );
  free ( dp );
  free ( dp_pre );
  free ( tmpbuf );
}
```

```c
/*******************************************************/
/*                                                     */
/*   2D Array fread() and fwrite() Program             */
/*   Sang Hwa  Lee,       1997.  8.  7.    in ATR      */
/*                                                     */
/*   Automatic Generation of Output Filename in 2Dfwrite   */
/*   Data Type : unsigned char                         */
/*                                                     */
/*   USAGE :  fread2d ( IMG, XSZ, YSZ, FILENAME )      */
/*              - IMG: unsigned char **image           */
/*              - FILENAME: char *input_filename       */
/*                                                     */
/*            fwrite ( IMG, XSZ, YSZ, FILENAME, TAG )  */
/*              - TAG: char *tag_name_of_ouputi,       */
/*                     "FILENAME_TAG" is output filename   */
/*                                                     */
/*******************************************************/


#include  <stdio.h>
#include  <stdlib.h>


/*------------------------------------------------------------------------------*/

fread_2D ( unsigned char **IMG, int XSZ, int YSZ, char *FILENAME )
{
  int            X, Y;
  unsigned char  *BUF;
  FILE           *FP;

  BUF = (unsigned char *)calloc( XSZ*YSZ, sizeof(unsigned char) );
  FP = fopen ( FILENAME, "rb" );
  fread( BUF, sizeof(unsigned char), XSZ*YSZ, FP );

  for ( Y=0 ; Y < YSZ ; Y++ )
    for ( X=0 ; X < XSZ ; X++ )
      IMG[Y][X] = BUF[XSZ*Y+X];

  free ( BUF );
  fclose ( FP );
}


/*------------------------------------------------------------------------------*/

fwrite_2D ( unsigned char **IMG, int XSZ, int YSZ, char *FILENAME, char *TAG )
{
  int            X, Y;
  unsigned char  *BUF;
  FILE           *FP;
  char           OUT_FILENAME[35];

  FP = fopen ( "__name", "w" );
  fprintf(FP, "%s_%s", FILENAME, TAG);
  fclose ( FP );

  FP = fopen ( "__name", "r" );
  fgets ( OUT_FILENAME, 34, FP );
  fclose ( FP );
  printf("%s", OUT_FILENAME);

  BUF = (unsigned char *)calloc( XSZ*YSZ, sizeof(unsigned char) );

  for ( Y=0 ; Y < YSZ ; Y++ )
    for ( X=0 ; X < XSZ ; X++ )
```

```
      BUF[XSZ*Y+X]   = IMG[Y+1][X+1];

  FP = fopen ( FILENAME, "wb" );
  fwrite( BUF, sizeof(unsigned char), XSZ*YSZ, FP );
  fclose ( FP );
  free ( BUF );
}

/*---------------------------------------------------------------------------*/
```

```
/*************************************************************************/
/*                                                                       */
/*  Memory Allocation Program for Image or Video                         */
/*  Sang Hwa Lee,    1997.  8.  7.     in  ATR                           */
/*                                                                       */
/*  2-dimensional, 3-Dimensional Allocation                              */
/*  type  :  int, float, unsigned char                                   */
/*  USAGE :  function_name(pointer, xsz, ysz, zsz)                       */
/*                                                                       */
/*  unsigned char **  Memory_2D_unsigned_char (int XSZ, int YSZ)         */
/*  unsigned char ***  Memory_3D_unsigned_char (int XSZ, int YSZ, int ZSZ) */
/*  int **  Memory_2D_int (int XSZ, int YSZ)                             */
/*  int ***  Memory_3D_int (int XSZ, int YSZ, int ZSZ)                   */
/*  float **  Memory_2D_float (int XSZ, int YSZ)                         */
/*  float ***  Memory_3D_float (int XSZ, int YSZ, int ZSZ)               */
/*                                                                       */
/*************************************************************************/


#include <stdio.h>
#include <stdlib.h>


/*-----------------------------------------------------------------------*/


unsigned char ** Memory_2D_unsigned_char (int XSZ, int YSZ)
{
  int              X, Y;
  unsigned char    **M;

  M = (unsigned char **)calloc(YSZ, sizeof(unsigned char *));

  for ( Y=0 ; Y < YSZ ; Y++ )
    M[Y] = (unsigned char *)calloc(XSZ, sizeof(unsigned char));
  return ( M );
}


/*-----------------------------------------------------------------------*/


unsigned char *** Memory_3D_unsigned_char (int XSZ, int YSZ, int ZSZ)
{
  int              X, Y, Z;
  unsigned char    ***M;

  M = (unsigned char ***)calloc(ZSZ, sizeof(unsigned char **));

  for ( Z=0 ; Z < ZSZ ; Z++ )
    M[Z] = (unsigned char **)calloc(YSZ, sizeof(unsigned char *));

  for ( Z=0 ; Z < ZSZ ; Z++ )
    for ( Y=0 ; Y < YSZ ; Y++ )
      M[Z][Y] = (unsigned char *)calloc(XSZ, sizeof(unsigned char));
  return ( M );
}


/*-----------------------------------------------------------------------*/


int ** Memory_2D_int (int XSZ, int YSZ)
{
  int              X, Y;
  int              **M;
```

```
  M = (int **)calloc(YSZ, sizeof(int *));

  for ( Y=0 ; Y < YSZ ; Y++ )
    M[Y] = (int *)calloc(XSZ, sizeof(int));
  return ( M );
}


/*----------------------------------------------------------------------------*/


int *** Memory_3D_int (int XSZ, int YSZ, int ZSZ)
{
  int              X, Y, Z;
  int              ***M;

  M = (int ***)calloc(ZSZ, sizeof(int **));

  for ( Z=0 ; Z < ZSZ ; Z++ )
    M[Z] = (int **)calloc(YSZ, sizeof(int *));

  for ( Z=0 ; Z < ZSZ ; Z++ )
    for ( Y=0 ; Y < YSZ ; Y++ )
      M[Z][Y] = (int *)calloc(XSZ, sizeof(int));
  return ( M );
}


/*----------------------------------------------------------------------------*/


float ** Memory_2D_float (int XSZ, int YSZ)
{
  int              X, Y;
  float            **M;

  M = (float **)calloc(YSZ, sizeof(float *));

  for ( Y=0 ; Y < YSZ ; Y++ )
    M[Y] = (float *)calloc(XSZ, sizeof(float));
  return ( M );
}


/*----------------------------------------------------------------------------*/


float *** Memory_3D_float (int XSZ, int YSZ, int ZSZ)
{
  int              X, Y, Z;
  float            ***M;

  M = (float ***)calloc(ZSZ, sizeof(float **));

  for ( Z=0 ; Z < ZSZ ; Z++ )
    M[Z] = (float **)calloc(YSZ, sizeof(float *));

  for ( Z=0 ; Z < ZSZ ; Z++ )
    for ( Y=0 ; Y < YSZ ; Y++ )
      M[Z][Y] = (float *)calloc(XSZ, sizeof(float));
  return ( M );
}


/*----------------------------------------------------------------------------*/
```