Bayesian Image Reconstruction Based on Voronoi Diagrams

G. F. Cabrera[1,2], S. Casassus[1]
and

N. Hitschfeld[2]
guille@das.uchile.cl

ABSTRACT

We present a Bayesian Voronoi image reconstruction technique (VIR) for interferometric data. Bayesian analysis applied to the inverse problem allows us to derive the *a-posteriori* probability of a novel parameterization of interferometric images. We use a variable Voronoi diagram as our model in place of the usual fixed pixel grid. A quantization of the intensity field allows us to calculate the likelihood function and *a-priori* probabilities. The Voronoi image is optimized including the number of polygons as free parameters. We apply our algorithm to deconvolve simulated interferometric data. Residuals, restored images and χ^2 values are used to compare our reconstructions with fixed grid models. VIR has the advantage of modeling the image with few parameters, obtaining a better image from a Bayesian point of view.

Subject headings: methods: data analysis — methods: numerical — methods: statistical — techniques: image processing — techniques: interferometric

1. Introduction

Astronomical interferometric data result from the addition of instrumental noise to the convolution of the sky image and the instrumental response. Because of incomplete sampling in the (u,v) plane, obtaining sky images from interferometric data is an instance of the inverse problem, and involves reconstruction algorithms.

The CLEAN method consists of modeling the side-lobe disturbances and subtract them iteratively from the dirty map (Högbom 1974). The CLEAN method works well for low noise and simple sources. But if the source has many complex features, or if the data is too noisy, CLEAN will do only a few iterations returning a noisy image (Högbom 1974). Another shortcoming is that CLEAN involves some ad-hoc parameters (the loop gain, stopping criteria, clean beam) that bias the final reconstruction, in the sense that CLEAN can give many different reconstructions for the same dataset.

The maximum entropy method (MEM) finds the image that simultaneously best fits the data, within the noise level, and maximizes the entropy S. This is done by minimizing

$$L_{\mathrm{MEM}}=\chi^2-\lambda S, \tag{1}$$

where, for the case of interferometric data, χ^2 can be calculated as

$$\chi^2=\sum_{k=1}^{N_{\mathrm{Vis}}}\frac{||V_k^{\mathrm{obs}}-V_k^{\mathrm{mod}}||^2}{\sigma_k^2}, \tag{2}$$

where the sum runs over all the N_{Vis} visibilities, the symbol $||z||$ stands for the modulus of the complex number z and σ_k is the root mean square (rms) noise of the corresponding visibility. λ is a control parameter and the entropy S

[1]Departamento de Astronomía, Universidad de Chile, Santiago, Casilla 36-D, Chile

[2]Departamento de Ciencias de la Computación, Universidad de Chile, Santiago

varies for different implementations (e.g. Narayan & Nityananda 1986). The entropy is used as a regularizing term in a degenerate inverse problem, when there are more free parameters than data. Different formulations for S appear in the literature. Some examples are $\sum_i \ln(I_i)$, $\sum_i I_i \ln(I_i)$, $\sum_i \ln(p_i)$, $\sum_i p_i \ln(p_i)$, where I_i is the specific intensity value at pixel i and $p_i = I_i / \sum_i I_i$ (see Piña & Puetter 1993, and references therein).

Cornwell & Evans (1985) used MEM in the AIPS VM task. Their method makes some approximations that diagonalize the Hessian matrix required to optimize their merit function. They used an entropy of the form $S = -\sum_i I_i \log(I_i/m_i)$, where the sum extends over all the pixels i, $\{I_i\}_{i=1}^{n}$ is the model image and $\{m_i\}_{i=1}^{n}$ is a prior image. However, the neglect of the side-lobe contribution to the Hessian may lead the optimization to local minima that still bear instrumental artifacts. Casassus et al. (2006)implemented a MEM algorithm based on the conjugate gradient method, without the use of the Cornwell and Evans approximation. They used an entropy of the form $S = -\sum_i I_i \log(I_i/M)$, where $\{I_i\}_{i=1}^{N}$ is the model image and M is a small intensity value, i.e they start with a blank image prior, and M is an intensity value much smaller than the noise.

Bayesian analysis is a powerful tool for image reconstruction techniques. In this application, our goal is to find the most probable image by maximizing its *a-posteriori* probability. For Bayesian methods, the *a-priori* and likelihood distributions are needed. To derive the *a-priori* probability the definition of an intensity quantum is needed. This quantum represents the minimum measurable intensity unit. The intensity in each pixel can be interpreted as a number of quanta $I_i = \sigma_q N_i$, where I_i is the intensity in pixel i, σ_q is the quantum size and N_i the number of quanta in pixel i.

Piña & Puetter (1993) used Bayesian analysis in the Pixon algorithm. They use a variable model and maximize $P(I,M|D)$, that is, the probability of the image I and model M given the data D. In their approach the model used to parameterize the image is a set of Gaussians which are used to average a pseudo-image. The pseudo-image starts as a maximum residual likelihood reconstruction and a local Gaussian pixon is assigned to each of its pixels. The number of pixons, and hence the number of free parameters, is reduced in each iteration.

Sutton & Wandelt (2006) have used Bayesian analysis for interferometric data, but using a fixed pixel grid to parameterize the model image. They use Gibbs sampling to determine the posterior density distribution.

The most typical model used in astronomy to represent the sky brightness distribution consists of a pixel grid. A big disadvantage of this grid is that the number of pixels remains fixed as well as their size. Often, uniform pixel grids involve more free parameters than really needed to fit the data.

The purpose of this paper is to explore Bayesian reconstruction with image models based on Voronoi tessellations in place of the usual pixelated image. We call this new deconvolution method "Voronoi image reconstruction" (VIR, hereafter). The advantage of using Voronoi models is that it is possible to use a smaller number of free parameters, as required by Bayesian theory. Our purpose is not optimal CPU efficiency; we search for the optimal image and model from a Bayesian point of view.

We used the Cosmic Background Imager (CBI, Padin et al. 2002) to illustrate our method. The CBI is a planar interferometer array with 13 antennas, each 0.9 m in diameter, mounted on a 6 m tracking platform. An example of CBI baselines is shown in Figure 1. The radius of the hole at the center of the (u,v) plane is the reciprocal of the minimum distance between two antennas, measured in wavelengths. The side-lobes of the CBI are caused mainly by this central hole in the (u,v) baselines.

We briefly summarize the elements of Bayesian theory that determine the probability distributions concerning our problem (Section 2). The new model based on Voronoi tessellations is described (Section 3), as well as optimization issues involved in our problem (Section 4). We discuss implementation details such as the optimal quantum size and number of Voronoi polygons (Section 5), compare reconstructions made with MEM and VIR (Section 6) and finally summarize our results (Section 7).

2. Bayesian Theory

An image model is required to parameterize the sky brightness distribution. The most typical model used in astronomy is a rectangular grid of uniform pixels. That configuration of pixels is

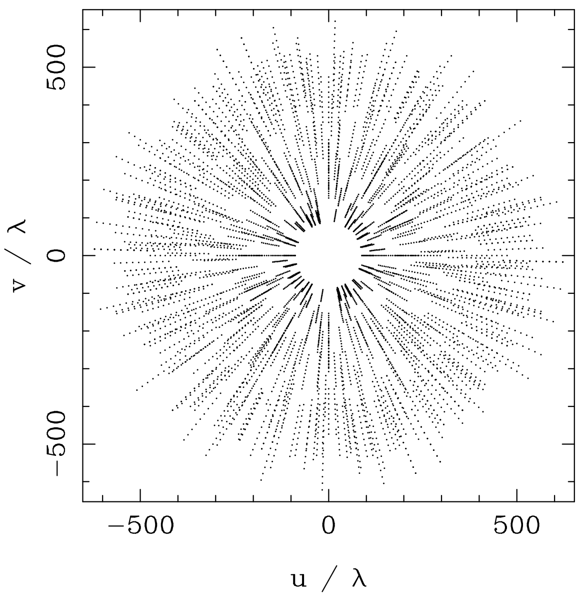

Fig. 1.— Coverage in the (u,v) plane of the CBI in the configuration used for our simulations.

the model M, and the distribution of brightness in the model is called an image I. We search for the image that represents as accurately as possible the visibility data D. The Bayesian image reconstruction approach, using a fixed model, tries to find the image that maximizes the probability $P(I|D,M)$, i.e. find the most probable image given the data and the model.

Using the Bayes theorem, we obtain

$$P(I|D,M)=\frac{P(D|I,M)P(I|M)}{P(D|M)}. \tag{3}$$

Since the data is fixed, $P(D|M)$ is a constant in the problem when the model is not considered as a variable. Thus, the fixed image model optimization problem reduces to

$$\max_I P(I|D,M)=\max_I P(D|I,M)P(I|M). \tag{4}$$

The first term, $P(D|I,M)$ is called the likelihood, and measures how well our data represents our image. The second term, $P(I|M)$ is called the image prior, and gives the *a-priori* probability of the image given the model, i.e. how probable is the image given only the model.

In the case of having a variable model, what we would like to find is the image and model that maximize $P(I,M|D)$, i.e. find the most probable image and model given the data. In this case we find

$$\begin{aligned} P(I,M|D) &= P(I|D,M)P(M|D) \\ &= \frac{P(D|I,M)P(I|M)P(M|D)}{P(D|M)} \\ &= \frac{P(D|I,M)P(I|M)P(M)}{P(D)}. \end{aligned} \tag{5}$$

Since the data is fixed, $P(D)$ is constant in our problem. As we cannot privilege one model over another in the absence of image and data, $P(M)$ is the same for all models, so it is not important for our analysis. This way, our optimization problem reduces to

$$\max_{I,M} P(I,M|D)=\max_{I,M} P(D|I,M)P(I|M). \tag{6}$$

2.1. Probability Distributions

Our data is a set of $N_{\rm Vis}$ observed visibilities $\{V_1^{\rm obs},V_2^{\rm obs},\cdots,V_{N_{\rm Vis}}^{\rm obs}\}$. If we have a certain model M and image I, we obtain model visibilities $\{V_k^{\rm mod}\}$ by simulating the interferometric observations over our image:

$$V_k^{\rm mod}=V^{\rm mod}(u_k,v_k) \tag{7}$$

$$=\int_{-\infty}^{+\infty} A(x,y)I(x,y)\exp[2\pi i(u_kx+v_ky)]\frac{dxdy}{\sqrt{1-x^2-y^2}}\ , \tag{8}$$

where $\{u_k,v_k\}$ are the coordinates of baseline k in the (u,v) plane and A is the primary beam. We thus have a set of $N_{\rm Vis}$ model visibilities. Assuming that each visibility is independent from the others and Gaussian noise, the likelihood is

$$P(D|I,M)=P(\{V_k^{\rm obs}\}_{k=1}^{N_{\rm Vis}}|\{V_k^{\rm mod}(I,M)\}_{k=1}^{N_{\rm Vis}}) \tag{9}$$

$$=\prod_{k=1}^{N_{\rm Vis}} P(V_k^{\rm obs}|V_k^{\rm mod})$$

$$=\prod_{k=1}^{N_{\rm Vis}} \frac{1}{2\pi\sigma_k^2}e^{-||V_k^{\rm obs}-V_k^{\rm mod}||^2/2\sigma_k^2}. \tag{10}$$

To obtain the image prior, $P(I|M)$, we calculate the statistical weight of a given distribution of counts (as in Piña & Puetter 1993; Sutton & Wandelt 2006). Consider a model consisting of n cells.

In the case of a traditional image, each pixel would be a cell. There is a number of N quanta falling into these cells. These are intensity quanta of some size $\sigma_{\rm q}$. In the case of a pixelated image, the intensity in each pixel i would be $I_i{=}\sigma_{\rm q}N_i$, where I_i is the intensity in cell i. Each quantum could fall into any of the n cells, so the total number of possible configuration for the N quanta will be n^N. The probability of the image given the model is the probability of a certain state $\{N_1,N_2,\cdots,N_n\}$ that represents that image, where N_i is the number of quanta in cell i. Consider a given image configuration defined by a particular distribution $\{N_i\}$. The image distribution is not changed in the $N!$ possible redistributions of counts between cells, provided each N_i is constant. The $\prod_i N_i!$ swaps of counts within each cell keep the same image configuration. The model M consists of the Voronoi diagram and the total number of quanta (i.e. n, the position of the generators and N), thus the *a-priori* probability is

$$P(I|M){=}P(\{N_i\}|n,N){=}\frac{N!}{n^N\prod_i N_i!}. \qquad (11)$$

As explained above, $\sigma_{\rm q}$ is an intensity quantum. It is also possible to describe the number of quanta per cell using a flux quantum $\sigma_i^{\rm F}$, where i is the index of the cell to which we associate the quantum. This flux quantum can be expressed in terms of the intensity quantum as $\sigma_i^{\rm F}{=}\sigma_{\rm q}A_i$, where A_i is the area of cell i. In this case, the number of quanta per cell is $N_i{=}F_i/\sigma_i^{\rm F}$, where $F_i{=}I_iA_i$ is the flux of cell i. This leads to $N_i{=}I_i/\sigma_{\rm q}$, which is the same expression for N_i obtained using the intensity quantum $\sigma_{\rm q}$. Using these cell-dependent flux quanta, the probability of a quantum falling into each cell will be $\frac{1}{n}$ for every cell, leaving the *a-priori* probability the same as Eq. 11.

2.2. MEM and Natural Entropy

In Bayesian theory, for a fixed model, the image I can be found by optimizing the *a-posteriori* probability:

$$\begin{aligned}\max_I P(I|D,M) &{=}\min_I(-\ln P(D|I,M)P(I|M))\\ &{=}\min_I\sum_{k=1}^{N_{\rm Vis}}\frac{||V_k^{\rm obs}-V_k^{\rm mod}||^2}{2\sigma_k^2}-\ln\left(\frac{N!}{n^N\prod_i N_i!}\right)\\ &{=}\min_I\frac{1}{2}\chi^2-S,\end{aligned} \qquad (12)$$

where we have defined the natural entropy $S{=}\ln\left(\frac{N!}{n^N\prod_i N_i!}\right)$. Sutton & Wandelt (2006) call the term $\ln(N!/\prod_i N_i!)$ the multiplicity prior. In the limit of large N_i,

$$S \simeq N\ln\frac{N}{n}-\sum_i N_i\ln N_i, \qquad (13)$$

and it can be seen that the Bayesian method is very similar to MEM in the sense that we are adjusting the image to the data while maximizing an entropy of the form of Eq. 13. VIR uses the natural entropy as a regularizing term.

3. A New Image Model based on Voronoi Diagrams

A Voronoi diagram is a division of the Euclidian plane into n regions $\mathcal{V}_i$ defined by n points $\vec{x_i}$ (called sites or generators) such that every coordinate $\vec{x}$ in the space belongs to $\mathcal{V}_i$ if and only if $||\vec{x}-\vec{x_i}||<||\vec{x}-\vec{x_j}||\ \forall\ j{\neq}i$. The result of the above definition is a set of polygons defined by the generators. Figure 2 shows an example of a Voronoi diagram. For further details on Voronoi diagrams see Okabe et al. (1992).

We propose a 2D Voronoi diagram in place of the usual pixelated, uniform grid, image as our model. We associate an intensity I_i to each of these polygons. The advantage of using a Voronoi diagram is that we can use just as many cells (i.e. free parameters) as the data requires. Our optimization parameters will be the position of each generator $\vec{x_i}{=}(x_i,y_i)$, and the intensity at each cell, I_i.

With our new model M consisting of n generators ($3{\times}n$ parameters, x_i,y_i and I_i for each generator), we can vary the number of free parameters as required by the optimization problem. We can see in equation (12) that the entropy S increases as the number of cell n decreases.

4. Optimization

The optimization problem can be seen as a maximization of the *a-posteriori* probability $\max_{I,M} P(I,M|D)$, or equivalently as a minimization of the more convenient merit function $L=\frac{1}{2}\chi^2-S$. The conjugate gradient method (CG) is often used for minimization problems where derivatives can be easily calculated. Though it is usually fast in convergence, CG has the problem of converging on local minima depending on the initial condition. The use of other optimization algorithms is postponed to future work.

The CG method searches parameters space using the gradient of the function to be minimized. The derivatives of this function are

$$\frac{\partial L}{\partial x}=\frac{1}{2}\frac{\partial \chi^2}{\partial x}-\frac{\partial S}{\partial x}, \tag{14}$$

$$\frac{\partial \chi^2}{\partial x}=2\sum_{k=1}^{N_{\rm Vis}}\frac{1}{\sigma_k^2}{\rm Re}\left((V_k^{\rm mod}-V_k^{\rm obs})^*\frac{\partial V_k^{\rm mod}}{\partial x}\right), \tag{15}$$

where x is any of the optimization parameters (x_i, y_i or I_i). The derivatives of the visibilities with respect to the position $\vec{x}_i=(x_i,y_i)$ of the i generator are

$$\frac{\partial V_k^{\rm mod}}{\partial x_i}=\sum_{j\in J_i}\Bigg[(I_i-I_j) \tag{16}$$

$$\times\sum_{l|\text{pixel } l\epsilon a_{ij}} A_l\Delta t_l(M_x t_l+b_x)e^{(t_l c_2+s_0 c_1)}\Bigg], \tag{17}$$

$$\frac{\partial V_k^{\rm mod}}{\partial y_i}=\sum_{j\in J_i}\Bigg[(I_i-I_j) \tag{18}$$

$$\times\sum_{l|\text{pixel } l\epsilon a_{ij}} A_l\Delta t_l(M_y t_l+b_y)e^{(t_l c_2+s_0 c_1)}\Bigg], \tag{19}$$

where I_i is the intensity in cell i, J_i is a set of the indices of the polygons adjacent to $\mathcal{V}_i$, a_{ij} is the edge which divides polygons $\mathcal{V}_i$ and $\mathcal{V}_j$, l sums over the pixels which intersect a_{ij}, A is the CBI primary beam. For further details see Sec. A.

The derivative of the visibilities with respect to the intensity of each cell I_i is

$$\frac{\partial V_k^{\rm mod}}{\partial I_i}=\frac{\sin(\pi u_k\Delta x)\sin(\pi v_k\Delta y)}{\pi^2 u_k v_k} \tag{20}$$

$$\times\sum_{\text{pixels } l\epsilon\mathcal{V}_i} A_l e^{2\pi i(u_k x_l+v_k y_l)}, \tag{21}$$

where $\vec{k}_k=(u_k,v_k)$ is the baseline corresponding to the pair of antennas k, Δx and Δy are the pixel width and height, and the sum extends over all the pixels inside $\mathcal{V}_i$.

The entropy only depends of the intensities I_i, so $\frac{\partial S}{\partial x_i}=\frac{\partial S}{\partial y_i}=0$, then (see Sec. A.2)

$$\frac{\partial S}{\partial I_i}=\frac{1}{\sigma_{\rm q}}(\sum_{k=1}^{N}\frac{1}{k}-\ln n-\sum_{k=1}^{N_i}\frac{1}{k}). \tag{22}$$

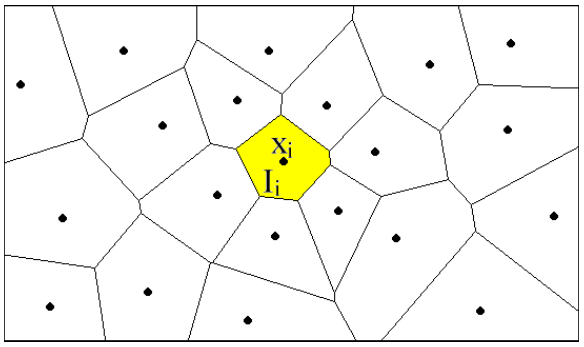

Fig. 2.— Example of Voronoi diagram.

5. VIR Design and Implementation

We have designed, and implemented in c++, VIR with 6 modules which include algorithms for:

- the generation of the Voronoi diagram
- calculation of model visibilities
- calculation of the merit function L to be optimized as well as its derivatives
- fitting a Voronoi diagram to an image

- the CG method
- the optimization of the number of polygons

VIR uses the CG method from Press et al. (1992) and searches for the position and intensities of the Voronoi polygons, x_i,y_i,I_i, that minimize our merit function L. The CG method modifies the intensities and also moves the positions of the Voronoi generators. This causes the shape of the Voronoi polygons to change as well. A general problem with CG is that it usually converges on local minima. For VIR in particular, though Voronoi polygons intensities adjust quite fine, the positions of the generators are difficult to modify substantially. The VIR parameter space is smooth enough in intensity space to converge to a good solution. But the parameter space in cell generator positions is very structured, and CG is quickly stuck on local minima.

Due to the fact that CG easily falls into local minima, we needed a good approximation for the initial Voronoi diagram. For this purpose we used a pixelated version of the Bayesian algorithm, where the model was a uniform grid. We decided to do a pure χ^2 (maximum likelihood, ML) reconstruction and use the fifth CG iteration as our starting image. We chose this particular iteration because on inspection the modeled images were still smooth. Pure χ^2 reaches convergence with noisy images, where the true image is unrecognizable. We then fitted a Voronoi diagram to the image (see Sec. B) and ran CG using the positions and intensities of the generators as our free parameters, which led to our final reconstruction. Truncation to a level of 10^{-5} quanta was used to enforce positivity.

An important issue to consider is the size of the quantum σ_q. Sutton & Wandelt (2006) treat σ_q as a free parameter. But, as we now explain, σ_q was held constant in this implementation of VIR. We treat the number of quanta per cell as a continuous variable in order to use the CG method. Entropy is maximized at σ_q=∞, where, for a given configuration of intensities $\{I_i\}$, N=0 and S=0. For every other value of N, the entropy will be negative. This means that even for large σ_q, the intensities I_i=$\sigma_\mathrm{q} N_i$ can have reasonable values (using small N_i). Figure 3 shows S as a function of N for 51 Voronoi generators and 3 different intensity distributions using the model tessellation of Figure 4c. We considered: 1- the VIR intensities of Figure 4c, 2- a uniform intensity distribution image (N_i=$\frac{N}{n}$ $\forall$ i), 3- a spike where all N are only in one cell (N_i=N, N_j=0 $\forall$ $j$$\neq$$i$). The curves of Figure 3 are obtained by keeping the intensities fixed and modifying σ_q in order to obtain different N. It can be seen on Figure 3 that the entropy is maximized at N=0, independently of the intensities $\{I_i\}$ of the model, where the optimal value of σ_q=∞ is achieved if the number of quanta per cell is treated as a continuous variable. If the number of quanta per cell were discrete variables, as in Sutton & Wandelt (2006), the choice of a big σ_q would admit only zero values for every cell. Otherwise, if one or more quanta fell in a given cell, the intensity of that cell would diverge as σ_q for arbitrarily large σ_q, causing a big χ^2 value. Therefore, in our continuous optimization the intensity quantum must be determined a-priori.

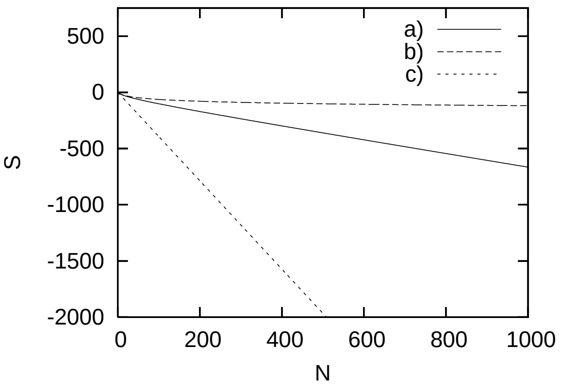

Fig. 3.— Entropy values for different N, n=51 and keeping $\{I_i\}$ fixed. This is achieved by varying σ_q. (a) VIR reconstruction intensities. (b) Uniform intensities distribution, N_i=$\frac{N}{n}$ $\forall$ i. (c) Only one cell has all the quanta. N_i=N, N_j=0 $\forall$ $j$$\neq$$i$.

In the Bayesian description of the entropy we count events that fall in each cell. It seems reasonable to take the noise level as the minimum value of intensity we can distinguish. So, σ_q should approximate the estimated thermal noise in the naturally weighted dirty map. The definition of the

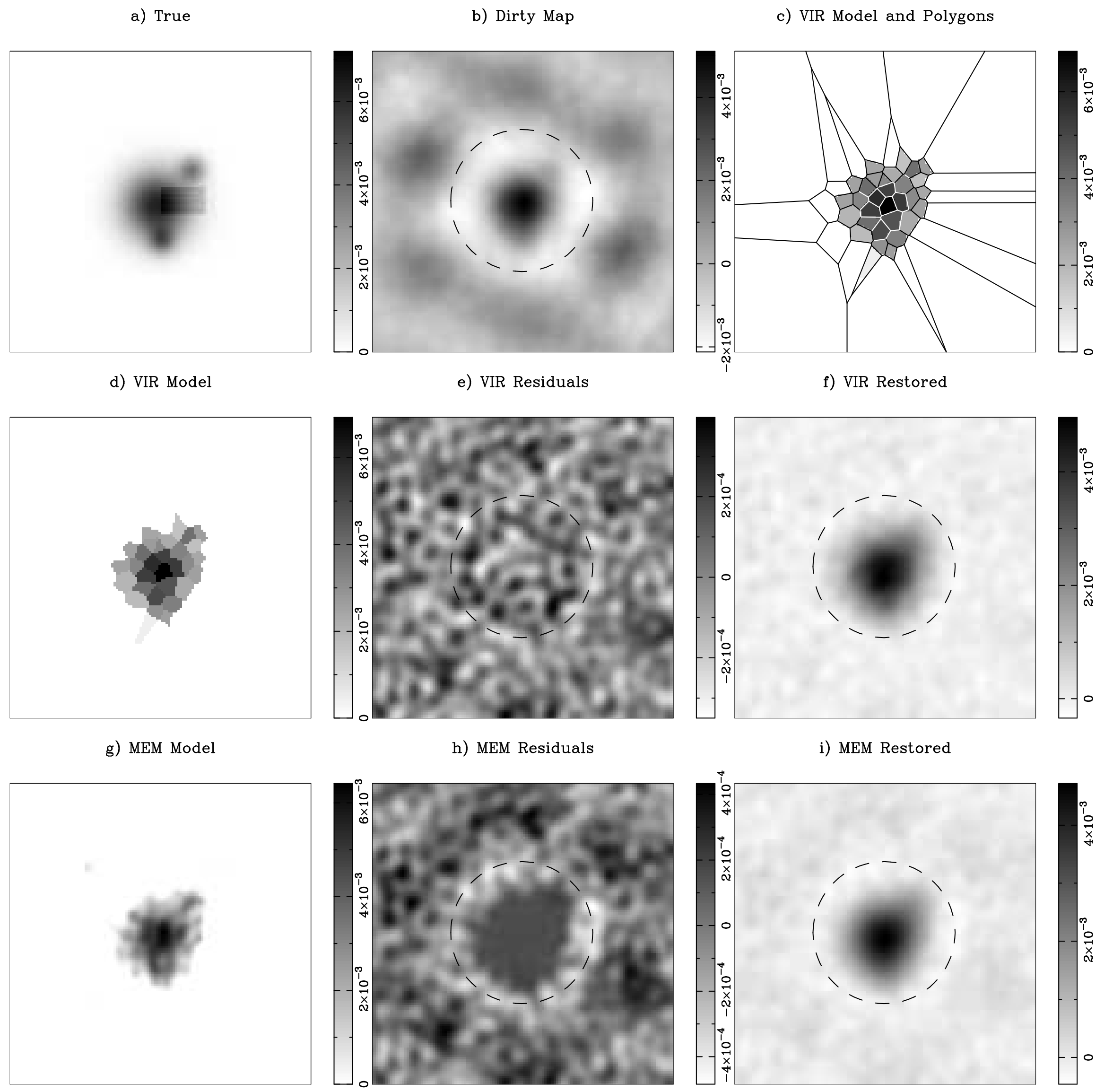

Fig. 4.— Comparison of MEM and VIR reconstruction techniques for a SNR of ∼52. (a) The true image. (b) Dirty map. (c) VIR reconstruction with its polygons drawn. (d) VIR reconstruction. (e) Dirty map of the VIR reconstruction residuals. (f) Restored image for the VIR model. (g) MEM reconstruction. (h) Dirty map of the MEM reconstruction residuals. (i) Restored image for the MEM model.

weighted dirty map (e.g. Briggs et al. 1999) is

$$I^{\mathrm{D}}(x,y) \equiv \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} W(u,v)V(u,v)e^{-2\pi i(ux+vy)}dudv, \tag{23}$$

$$W(u,v) = \frac{1}{\sum_k w_k}\sum_k w_k \delta(u-u_k, v-v_k), \tag{24}$$

where the sums extend over all visibilities, w_k are the weights given to visibility k and δ is the two-dimensional Dirac delta function. Propagating the thermal noise, we get for the standard deviation of the dirty map

$$\sigma^{\mathrm{D}}_{\mathrm{rms}} = \sqrt{\frac{\sum_k w_k^2 \sigma_k^2}{(\sum_k w_k)^2}}, \tag{25}$$

where σ_k are the visibilities standard deviations. To take into account model pixels correlated by the interferometer beam, we should multiply the previous expression by $\sqrt{N_{\mathrm{beam}}}$, where N_{beam} is the number of pixels inside a beam pattern. This leads to

$$\sigma_{\mathrm{rms}} = \sqrt{\frac{\sum_k w_k^2 \sigma_k^2}{(\sum_k w_k)^2}}\sqrt{N_{\mathrm{beam}}}. \tag{26}$$

For natural weights, $\sigma_k^2 = \frac{1}{w_k}$,

$$\sigma_{\mathrm{rms}} = \sqrt{\frac{N_{\mathrm{beam}}}{\sum_k w_k}} = \sqrt{\frac{N_{\mathrm{beam}}}{\sum_k \frac{1}{\sigma_k^2}}}. \tag{27}$$

We calculated the noise with natural weighting, $w_k = \frac{1}{\sigma_k^2}$, because this is the weight we give to each individual visibility data in the optimization of the merit function.

Once we have the value of σ_{q} we search for the optimal number of cells n. In Figure 5 we plot the optimal merit function for different n and σ_{q}. These reconstructions were made over a simulation of CBI observations on a mock sky image (Figure 4a). We averaged over 100 reconstructions with different realizations of Gaussian noise. The average curves shown in Figure 5, start with n=10 and end with n=100 for even n. One single reconstruction for all n took about two hours using an AMD Athlon64 XP3000 processor with 1GB of DDR RAM at 333 MHz, so the 300 reconstructions took about 25 CPU days, but we distributed the work in six computers, so it took about 5 real days in total. It can be seen that for a signal to noise ratio (SNR) of ∼52, on average, the optimal number of polygons n is between 50 and 55. When σ_{q} is diminished to $\frac{1}{10}\sigma_{\mathrm{q}}$, on average, the optimal merit function is found at n close to 30. For σ_{q}=10σ_{q}, the optimal n is found between 80 and 90. It can be seen that as we increase the value of σ_{q} we reach lower values for our function, as discussed above. Furthermore, the optimal number of polygons increases.

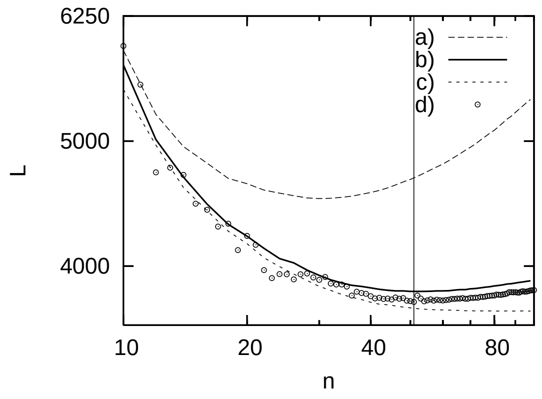

Fig. 5.— The merit function L for different σ_{q} and number of polygons n. The lines are averages taken over 100 different realizations of noise for each n. (a) Reconstructions made using σ_{q}= $\frac{1}{10}\sigma_{\mathrm{rms}}$. (b) Reconstructions made using σ_{q}=σ_{rms}. (c) Reconstructions made using σ_{q}=10σ_{rms}. (d) L as a function of n for a practical application of VIR to the simulated visibilities used in the reconstructions of Figure 4. In this practical application, the minimum L was found at n=51, and is indicated by a vertical line.

6. Example Reconstruction

6.1. Mock Dataset

The mock sky image we used for simulations is a 256×256 image consisting of three Gaussians and a rectangle. Figure 4a shows this image on a 128×128 pixel field. Pixels are 0.75′×0.75′, while the CBI's primary beam is of 45′ FWHM (60 pixels), so most of the emission lies under the beam. We simulated a CBI observation of 3620 visibilities over this image and added Gaussian noise to the visibilities in order to reach a SNR of ∼52. This

SNR was calculated by taking the maximum intensity from the dirty map using natural weights, and using the noise $\sigma_{\rm rms}^{\rm D}$ (see Eq. 25). Simulation of the CBI observations is performed with the MockCBI program (Pearson 2000, private communication), which calculates the visibilities $V(u,v)$ on the input images $I(x,y)$ with the same uv sampling as a reference visibility dataset (Eq. 8). Thus MockCBI creates the visibility dataset that would have been obtained had the sky emission followed the true image. Figure 4b shows the dirty map calculated over these simulated visibilities using the DIFMAP package (Shepherd 1997). The CBI's primary beam is drawn as a dashed circle. The secondary side-lobes due to the central discontinuity in u-v coverage can be distinguished in Figure 4b at a level comparable to the true emission.

6.2. MEM Reconstruction

The VIR method was compared with the MEM algorithm described in Casassus et al. (2006). To fit the model image to the observed visibilities, MEM calculates the model visibilities required by its merit function $L_{\rm MEM}$. The model visibilities are those obtained by a simulation of CBI observations had the sky followed the model image . The free-parameters of our MEM model are the pixels in the model 64×64 image. The model functional we minimize is $L_{\rm MEM}{=}\chi^2{-}\lambda S$, with the entropy $S{=}{-}\sum_i I_i \log I_i/M$, where M is a default pixel value well below the noise level, and $\{I_i\}_{i=1}^{N}$ is the model image. We started with the fifth iteration of a pure χ^2 reconstruction ($\lambda{=}0$) as initial condition for the CG minimization. This is the same ML initial condition used in our VIR method. Figure 4g shows the reconstructed image using $\lambda{=}\frac{100}{\sigma_{\rm rms}}$ and $M{=}10^{-2}\sigma_{\rm rms}$ inset on a larger 128×128 image [1].

6.3. VIR Reconstruction

The MEM algorithm described above requires the prior assignment of the λ and M parameters as well as the entropy formula. In contrast, our VIR algorithm is free from such arbitrary parameters (provided the optimal $\sigma_{\rm q}$ is indeed equal to $\sigma_{\rm rms}$). For our VIR method, we only need to find the number of polygons to be used. In order to find the optimal number of polygons we reconstructed with different numbers of generators in a range covering each natural number from $n{=}6$ to $n{=}100$. We found a minimum at $n{=}51$. Figure 5 summarizes this search. The whole search for a particular realization of noise took about 10 hours on the AMD Athlon64 XP3000 processor with 1GB of DDR RAM at 333 MHz. The VIR reconstruction using 51 polygons is shown in Figure 4c, where the Voronoi cells have also been drawn. Figure 4d shows the same model but without drawing the Voronoi mesh.

6.4. Results

The quality of each reconstruction can be assessed by visual inspection, comparing the VIR and MEM model images with the true image. The MEM model looks similar to the true image but is noisy. The density of Voronoi generators in the VIR model is greater where there is more emission in the true image, approximating the true image with only a few polygons. We calculated $\chi^2_{\rm im}{=}\sum_i (I_i^{\rm mod}{-}I_i^{\rm true})^2$, where $I_i^{\rm mod}$ is the intensity at pixel i of the model image (MEM or VIR), $I_i^{\rm true}$ is the intensity at pixel i of the true image, and the sum extends over all pixels in the images. $\chi^2_{\rm im}$ gives a measure of how well the model fits the true image. It can be seen in Table 1 that the VIR reconstruction has a better $\chi^2_{\rm im}$ than MEM, showing that the VIR model is closer to the true image than the MEM model.

Figures 4e and 4h show the VIR and MEM models residuals. Residual images are the dirty map of the residuals of the visibilities, calculated over the optimal model visibilities. It can be noted on Figure 4e that the VIR residuals are very good, showing only noise. On the other hand, in the MEM residals (Figure 4h) the object shape can clearly be distinguished as well as the CBI's side-lobes. The object seems to be more compact in the model than in its MEM residuals; as expected these residuals are convolved with the synthetic beam.

Restored images are shown in Figures 4f and 4i. These images are obtained by convolving the models with a Gaussian point spread function (PSF) given by DIFMAP and adding the dirty map of the residuals visibilities. On Figures 4f and 4i it can be assessed that VIR produces improved restored

[1] We choose to display the sky images in a larger field than the domain of free parameters; larger fields are required to highlight secondary side-lobes

Table 1
Comparison between MEM and VIR reconstructions.

	χ^2	$\frac{\chi^2}{n_{\rm data}}$	L	$\chi^2_{\rm im}$
MEM	7354.85	1.016	12192.6	0.001608
VIR	7221.04	0.997	3753.28	0.001396

images relative to MEM. The VIR restored image is similar to that expected given the instrumental noise: it approximates the true image convolved with a Gaussian PSF plus a uniform noise level. In the MEM restored image, on the other hand, the CBI side-lobes can still be distinguished.

The number of optimization parameters in MEM are 64×64=4096, while the VIR method has only 51 triplets (cell's (x,y) position and intensity) i.e. 153 free parameters. This smaller number of parameters causes the Bayesian entropy to be greater than the pixelated version, obtaining a smaller value for our merit function L to be minimized.

Table 1 also shows $\frac{\chi^2}{n_{\rm data}}$ values, where $n_{\rm data}$ is the number of data points (3620×2 in our case). A good reconstruction should have a $\frac{\chi^2}{n_{\rm data}}$ value close to 1. It can be seen that the VIR model gives a value of $\frac{\chi^2}{n_{\rm data}}$ closer to 1 than the MEM reconstruction.

7. Conclusions

We have introduced a Bayesian Voronoi image reconstruction (VIR) technique for interferometric data where the image is represented by a Voronoi tessellation in place of the usual pixelated image. The advantage of Voronoi models is that we can use a smaller number of free parameters, as required by the Bayesian analysis of a discretized intensity field. Our purpose is not optimal CPU efficiency; we search for the optimal image and model from a Bayesian point of view. The free parameters of our model are the Voronoi generators positions (x_i,y_i) and intensities I_i. The following points summarize our work:

- We discretized the intensity field in order to calculate *a priori* probabilities. We defined a quantum intensity value $\sigma_{\rm q}$ such that $I_i = \sigma_{\rm q} N_i$, where I_i is the intensity at cell i and N_i the number of quanta in cell i.

- We calculated the analytical derivatives required by the conjugate gradient and cross checked them by finite differences. Because the parameter space in cell generators positions is very structured, the positions of the Voronoi generators are difficult to change. As initial condition we took a Voronoi tessellation adjusted to an interrupted maximum likelihood reconstruction.

- We simulated a CBI observation over a true image and reconstructed sky images from this mock visibility dataset using MEM and VIR.

- We defined the value of $\sigma_{\rm q}$ as the estimated noise of the dirty map and searched for the optimal number of Voronoi polygons for our example dataset.

- We finally compared the MEM and VIR models, residuals and restored images. The VIR model is closer to our true image than the MEM model. Residuals and restored images are also better in VIR than in MEM. We found that VIR model visibilities give a better fit to the data than MEM, in the sense that χ^2 is closer to its expected value.

We are grateful to Tim Pearson for advice on FFTs and the use of MOCKCBI. G.F.C. and S.C. acknowledge support from FONDECYT grant 1060827, and from the Chilean Center for Astrophysics FONDAP 15010003.

A. Derivatives

Our merit function for minimization is

$$L = \frac{1}{2}\sum_{j=1}^{N_{\rm Vis}} \frac{||V_j^{\rm mod} - V_j^{\rm obs}||^2}{\sigma_j^2} - \ln\left(\frac{N!}{n^N \prod_{i=1}^{n} N_i!}\right) \tag{A1}$$

$$= \frac{1}{2}\chi^2 - S. \tag{A2}$$

So, the derivative of L with respect to any variable x is

$$\frac{\partial L}{\partial x} = \frac{1}{2}\frac{\partial \chi^2}{\partial x} - \frac{\partial S}{\partial x} \tag{A3}$$

A.1. Calculation of the Derivatives of χ^2

χ^2 derivatives with respect to any variable x can be obtain as follows

$$\frac{\partial}{\partial x}\frac{1}{2}\chi^2 = \frac{\partial}{\partial x}\left(\frac{1}{2}\sum_{k=1}^{N_{\rm Vis}} \frac{||V_k^{\rm mod} - V_k^{\rm obs}||^2}{\sigma_k^2}\right)$$
$$= \sum_{k=1}^{N_{\rm Vis}} \frac{1}{\sigma_k^2}\left({\rm Re}(V_k^{\rm mod} - V_k^{\rm obs}){\rm Re}\left(\frac{\partial V_k^{\rm mod}}{\partial x}\right) + {\rm Im}(V_k^{\rm mod} - V_k^{\rm obs}){\rm Im}\left(\frac{\partial V_k^{\rm mod}}{\partial x}\right)\right), \tag{A4}$$

where its necessary to calculate the model visibilities derivatives with respect to x.

A.1.1. Calculation of $\frac{\partial V_k^{\rm mod}}{\partial I_i}$

In our Voronoi tessellation representation of the sky image

$$V(\vec{k}) = \sum_{i}^{N_{\rm V}} I_i \int_{\mathcal{V}_i} A(\vec{x}) e^{2\pi i \vec{k}\vec{x}} d\vec{x}, \tag{A5}$$

where $N_{\rm V}$ is the number of polygons, $\mathcal{V}_i$ is polygon i and I_i its intensity. We neglected the $\sqrt{1-x^2-y^2}$ term which is close to 1, but it can easily be included in $A(\vec{x})$. After derivation and defining $f_k(\vec{x}) \equiv A(\vec{x}) e^{2\pi i \vec{k}_k \vec{x}}$ we obtain

$$\frac{\partial V_k^{\rm mod}}{\partial I_i} = \int\int_{\mathcal{V}_i} f_k(\vec{x}) d^2x, \tag{A6}$$

$$= \frac{\sin(\pi u_k \Delta x)\sin(\pi v_k \Delta y)}{\pi^2 u_k v_k} \sum_{\text{pixels } l \epsilon \mathcal{V}_i} A_l e^{2\pi i (u_k x_l + v_k y_l)}, \tag{A7}$$

$$\simeq \Delta x \Delta y \sum_{\text{pixels } l \epsilon \mathcal{V}_i} A_l e^{2\pi i (u_k x_l + v_k y_l)} \tag{A8}$$

for small Δx and Δy.

A.1.2. Calculation of $\frac{\partial V_k^{\rm mod}}{\partial x_i}$ *and* $\frac{\partial V_k^{\rm mod}}{\partial y_i}$

To evaluate $\frac{\partial V_k}{\partial x_i}$ we move the generator $\vec{x}_i$ an infinitesimal quantity δ_x parallel to the $\hat{x}$ axis as in Figure 6. We will calculate

$$\frac{\partial V_k}{\partial x_i} = \lim_{\delta_x \to 0} \frac{\Delta V}{\delta_x}, \tag{A9}$$

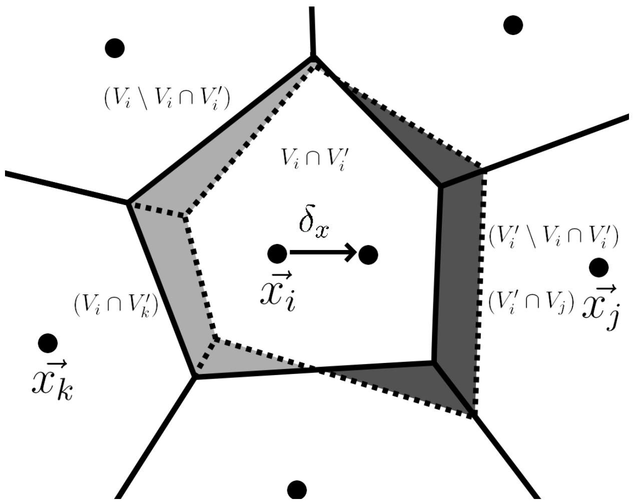

Fig. 6.— Voronoi tessellation before and after translating the site $\vec{x}_i$ by $\vec{\delta_x}$. Voronoi generators are represented with dots. The solid lines are the polygons before moving $\vec{x_i}$. The dotted lines represent the new polygons after varying $\vec{x_i}$.

where $\Delta V_k = V_k(\vec{x}_1,\cdots,\vec{x}_i+\vec{\delta_x},\cdots,\vec{x}_{N_V}) - V_k(\vec{x}_1,\cdots,\vec{x}_i,\cdots,\vec{x}_{N_V})$.

It can be seen in Figure 6 that when moving the generator $\vec{x}_i$, the only polygons modified are $\mathcal{V}_i$ and its neighbors. Using this, Eq. A5 leads to

$$\begin{aligned} \Delta V_k &= I_i' \int_{\mathcal{V}_i'} f_k(\vec{x}) d\vec{x} - I_i \int_{\mathcal{V}_i} f_k(\vec{x}) d\vec{x} \\ &\quad + \sum_{j \in J_i} \left(I_j' \int_{\mathcal{V}_j'} f_k(\vec{x}) d\vec{x} - I_j \int_{\mathcal{V}_j} f_k(\vec{x}) d\vec{x} \right), \end{aligned} \tag{A10}$$

where $\mathcal{V}_i$ is the polygon generated by $\vec{x}_i$ before moving, $\mathcal{V}_i'$ is the same polygon after moving $\vec{x}_i$, J_i is the set of indices of the polygons that are neighbors to $\mathcal{V}_i$ and J_i' is the set of indices of the polygons that are neighbors to $\mathcal{V}_i'$.

It can be seen in Figure 6 that

$$\mathcal{V}_i = (\mathcal{V}_i \cap \mathcal{V}_i') \cup (\mathcal{V}_i \setminus \mathcal{V}_i \cap \mathcal{V}_i'), \quad \mathcal{V}_i' = (\mathcal{V}_i \cap \mathcal{V}_i') \cup (\mathcal{V}_i' \setminus \mathcal{V}_i \cap \mathcal{V}_i'), \tag{A11}$$

$$\mathcal{V}_j = (\mathcal{V}_j \cap \mathcal{V}_j') \cup (\mathcal{V}_i' \cap \mathcal{V}_j), \quad \mathcal{V}_j' = (\mathcal{V}_j \cap \mathcal{V}_j') \cup (\mathcal{V}_i \cap \mathcal{V}_j'), \tag{A12}$$

so, Eq. A10 is

$$\begin{aligned} \Delta V_k &= (I_i' - I_i) \int_{\mathcal{V}_i' \cap \mathcal{V}_i} f_k(\vec{x}) d\vec{x} \\ &\quad + \sum_{j \in J_i} \left[(I_i' - I_j) \int_{\mathcal{V}_i' \cap \mathcal{V}_j} f_k(\vec{x}) d\vec{x} + (I_j' - I_i) \int_{\mathcal{V}_i \cap \mathcal{V}_j'} f_k(\vec{x}) d\vec{x} \right]. \end{aligned} \tag{A13}$$

In our case the cells' intensities don't depend of the position of the generators, so we obtain

$$\Delta V_k = \sum_{j \in J_i} \left[(I_i - I_j) \left(\int_{\mathcal{V}_i' \cap \mathcal{V}_j} f_k(\vec{x}) d\vec{x} - \int_{\mathcal{V}_i \cap \mathcal{V}_j'} f_k(\vec{x}) d\vec{x} \right) \right]. \tag{A14}$$

It can be seen in Figure 6 that to obtain ΔV_k we must integrate only over the shaded regions. For this purpose, for each region between $\vec{x_i}$ and $\vec{x_j}$ we will define a coordinate system

$$\hat{s}{=}{-}\cos\alpha_j\hat{x}{+}\sin\alpha_j\hat{y}. \quad \hat{t}{=}\sin\alpha_j\hat{x}{+}\cos\alpha_j\hat{y}, \tag{A15}$$

where α_j is the angle formed by the $-\hat{x}$ axis and the edge a_{ij} between $\vec{x_i}$ and $\vec{x_j}$ (see Figure 7). Using this change of coordinates, the integral over the region of interest is

$$\int_{\mathcal{V}_i'\cap\mathcal{V}_j} f_k(x,y)dxdy{=}\int_{\mathcal{V}_i'\cap\mathcal{V}_j} f_k(s,t)dsdt. \tag{A16}$$

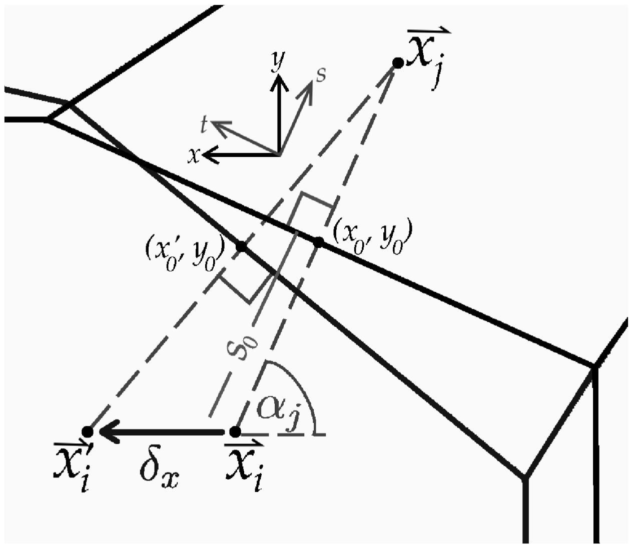

Fig. 7.— Change of coordinates from (x,y) to (s,t).

Let $\vec{x_i}{=}(x_i,y_i)$ be the position of the i cell's generator, $\vec{x_j}{=}(x_j,y_j)$ one of its neighbor, and $\vec{x_i}'{=}(x_i{+}\delta_x,y_i)$ the site's position after moving it a quantity δ_x. We define $\vec{x_0}{\equiv}(x_0,y_0)$ as the point in the intersection of the segment formed by $\vec{x_i}$ and $\vec{x_j}$ and its respective edge a_{ij}. The same way, we define $\vec{x_0}'{=}(x_0',y_0')$ as the point in the intersection of the segment formed by $\vec{x_i}'$ and $\vec{x_j}$ and its respective edge a_{ij}'. It can be seen on Figure 7 that $x_0{=}\frac{x_i+x_j}{2}$, $x_0'{=}x_0{+}\frac{\delta}{2}$ and $y_0'{=}y_0{=}\frac{y_i+y_j}{2}$.

The edge a_{ij} is defined in the new coordinate system by

$$s{=}s_0{=}{-}x_0\cos\alpha_j{+}y_0\sin\alpha_j. \tag{A17}$$

In the same way, the edge a_{ij}' is defined in the original coordinate system by

$$y{=}m(x{-}x_0'){+}y_0, \tag{A18}$$

where

$$m{\equiv}\frac{x_i{+}\delta_x{-}x_j}{y_j{-}y_i}. \tag{A19}$$

We can define the same line in our new coordinate system as

$$s{=}m't{+}b', \tag{A20}$$

where

$$m' \equiv -\frac{\cos\alpha_j + m\sin\alpha_j}{\sin\alpha_j - m\cos\alpha_j}, \tag{A21}$$

$$b' \equiv \frac{-mx'_0 + y_0}{\sin\alpha_j - m\cos\alpha_j}. \tag{A22}$$

This can be approximated to first order in δ_x as

$$m' \simeq \delta_x M_x, \tag{A23}$$

$$b' \simeq s_0 + \delta_x B_x, \tag{A24}$$

where

$$M_x \equiv \frac{\sin^2\alpha_j}{y_j - y_i} = \frac{\sin\alpha_j\cos\alpha_j}{x_i - x_j}, \tag{A25}$$

$$B_x \equiv \frac{\sin\alpha_j}{y_j - y_i}(s_0\cos\alpha_j + x_i) = \frac{\cos\alpha_j}{x_i - x_j}(s_0\cos\alpha_j + x_i). \tag{A26}$$

The integral in Eq. A14 using our new coordinate system will be

$$\mathcal{I} = \int_{\mathcal{V}'_i \cap \mathcal{V}_j} f_k(\vec{x}) d\vec{x} - \int_{\mathcal{V}_i \cap \mathcal{V}'_j} f_k(\vec{x}) d\vec{x} \tag{A27}$$

$$= \int\int_{a_{ij}}^{a'_{ij}} A(\vec{x}) e^{2\pi i(ux+vy)} dx dy. \tag{A28}$$

If we use $A(\vec{x})$ in the (s,t) coordinate system as a pixelated image, Eq. A28 will be

$$\mathcal{I} = \sum_{l \,\epsilon\, \text{pixeles de } a_i} A_l \int_{t^1_{ijl}}^{t^2_{ijl}} \int_{s_0}^{m't+b'} e^{2\pi i(ux(s,t)+vy(s,t))} ds dt, \tag{A29}$$

where t^1_{ijl} and t^2_{ijl} are the t coordinate of the beginning and end of the portion of the edge a_{ij} that intersects pixel l. Developing the previous expression,

$$\mathcal{I} = \sum_l A_l \int_{t^1_{ijl}}^{t^2_{ijl}} \int_{s_0}^{m't+b'} e^{2\pi i(u(-s\cos\alpha_j + t\sin\alpha_j) + v(s\sin\alpha_j + t\cos\alpha_j))} ds dt, \tag{A30}$$

$$\simeq \sum_l \frac{A_l}{\pi c_2} e^{2\pi i(s_0 c_1 + \bar{t}_{ijl} c_2)} \kappa_{ijl} \delta_x, \tag{A31}$$

where we defined

$$c_1 \equiv -u\cos\alpha_j + v\sin\alpha_j, \tag{A32}$$

$$c_2 \equiv u\sin\alpha_j + v\cos\alpha_j, \tag{A33}$$

$$\kappa_{ijl} \equiv (M_x\bar{t}_{ijl} + B_x)\sin(\pi c_2 \Delta t_{ijl}) + i\frac{M_x}{2}\left(\frac{\sin(\pi c_2 \Delta t_{ijl})}{\pi c_2} - \Delta t_{ijl}\cos(\pi c_2 \Delta t_{ijl})\right), \tag{A34}$$

$$\bar{t}_{ijl} \equiv \frac{t^1_{ijl} + t^2_{ijl}}{2}, \tag{A35}$$

$$\Delta t_{ijl} \equiv \frac{t^2_{ijl} - t^1_{ijl}}{2}. \tag{A36}$$

In the calculation above we integrated over the fraction of the edge that falls inside pixel l and then summed these integrals over the whole edge a_i. It is also possible to approximate the integral of Eq. A30 as $\int_{t^1_{ijl}}^{t^2_{ijl}} g(t)dt = g(\bar{t}_{ijl})\Delta t_{ijl}$, which is equivalent to taking the limit over the integral $\mathcal{I}$ of Eq. A31, $\lim_{\Delta t_{ijl}\to 0}\mathcal{I}$, obtaining

$$\mathcal{I}=\sum_l A_l \Delta t_{ijl}(M_x \bar{t}_{ijl}+B_x)e^{2\pi i(\bar{t}_{ijl}c_2+s_0c_1)}\delta_x. \tag{A37}$$

We found by direct evaluation that the difference between Eq. A37 and Eq. A31 is negligible, so, for simplicity, we will use Eq. A37. Introducing Eq. A37 in Eq. A14, we obtain

$$\Delta V_k=\delta_x \sum_{j\in J_i}\left[(I_i-I_j)\sum_l A_l \Delta t_{ijl}(M_x \bar{t}_{ijl}+B_x)e^{2\pi i(\bar{t}_{ijl}c_2+s_0c_1)}\right], \tag{A38}$$

so, according to Eq. A9, the derivative of the k visibility with respect to the position x of polygon i is

$$\frac{\partial V_k}{\partial x_i} = \lim_{\delta_x\to 0}\frac{\Delta V}{\delta_x}, \tag{A39}$$

$$= \sum_{j\in J_i}\left[(I_i-I_j)\sum_l A_l \Delta t_{ijl}(M_x \bar{t}_{ijl}+B_x)e^{2\pi i(\bar{t}_{ijl}c_2+s_0c_1)}\right]. \tag{A40}$$

Similarly, for the derivative with respect to the position y of the i polygon we obtain

$$\frac{\partial V_k}{\partial y_i}=\sum_{j\in J_i}\left[(I_i-I_j)\sum_l A_l \Delta t_{ijl}(M_y \bar{t}_{ijl}+B_y)e^{2\pi i(\bar{t}_{ijl}c_2+s_0c_1)}\right], \tag{A41}$$

where

$$M_y \equiv \frac{\cos^2\alpha_j}{x_i-x_j}=\frac{\sin\alpha_j\cos\alpha_j}{y_j-y_i}, \tag{A42}$$

$$B_y \equiv \frac{\sin\alpha_j}{y_j-y_i}(s_0\sin\alpha_j-y_i)=\frac{\cos\alpha_j}{x_i-x_j}(s_0\sin\alpha_j-y_i). \tag{A43}$$

A.2. Calculation of the Derivatives of S

We defined our entropy as

$$S = \ln\left(\frac{N!}{n^N\prod_{i=1}^n N_i!}\right) \tag{A44}$$

$$= \ln(N!)-N\ln(n)-\sum_{i=1}^n \ln(N_i!) \tag{A45}$$

$$= \ln\Big(\Gamma(N+1)\Big)-N\ln(n)-\sum_{i=1}^n \ln\Big(\Gamma(N_i+1)\Big), \tag{A46}$$

where $N_i=\frac{I_i}{\sigma_q}$ is the number of quanta in cell i, $N=\sum_i N_i$ and Γ is the Gamma function. It can be seen that this function does not depend on the position of the Voronoi generators, so

$$\frac{\partial S}{\partial x_i}=\frac{\partial S}{\partial y_i}=0. \tag{A47}$$

Using Weierstrass' definition of the Gamma function

$$\Gamma(z)=z^{-1}e^{-\gamma z}\prod_{n=1}^{\infty}\left[\left(1+\frac{z}{n}\right)^{-1}e^{z/n}\right],$$

where γ is Euler's constant, we can obtain

$$\frac{\partial \ln\Big(\Gamma(z{+}1)\Big)}{\partial z}{=}{-}\gamma{+}\sum_{n=1}^{z}\frac{1}{n} \tag{A48}$$

so, the derivative of S with respect to I_i is

$$\frac{\partial S}{\partial I_i}{=}\frac{1}{\sigma_{\mathrm{q}}}(\sum_{k=1}^{N}\frac{1}{k}{-}\ln n{-}\sum_{k=1}^{N_i}\frac{1}{k}). \tag{A49}$$

A.3. Finite Difference Cross Check on the Derivatives

Numerical calculation of the derivatives by finite differences is not very accurate, in particular for the position of the generators. Finite difference derivatives are calculated as $\frac{\partial L}{\partial x}{=}\frac{L(x+\delta)-L(x)}{\delta}$, where δ is a small displacement of x. In the case of the positions of the generators, if δ is too small, the pixelization of the Voronoi diagram (needed to obtain the model visibilities) will not change after the displacement δ. On the other hand, if δ is too big, the generator displacement may cause the function to change abruptly, as explained below. That is why we calculated the analytical expression for the derivatives.

To verify that our derivatives are correctly calculated and programmed, we compared our analytical result with a numerical calculation. We created a Voronoi tessellation of 50 polygons with random positions and intensities and calculated the analytical and numerical derivatives using these parameters $\{x_i,y_i,I_i\}$. For the case of $\frac{\partial L}{\partial x_i}$ and $\frac{\partial L}{\partial y_i}$ this numerical cross check consists of moving each Voronoi generator a quantity δ from -0.1 to 0.1 with an interval of 10^{-3} in units of the total size of the square image. We evaluate the merit function L at each position intervals, thus obtaining two sequences $\{L_i\}_{i=1}^{2\times10^2}$. We then fitted a polynomial of order four to the curve defined by each sequence $\{L_i\}$ and calculated the derivative of the polynomial at δ=0. For the case of $\frac{\partial L}{\partial I_i}$ we varied the intensity of cell i from $-\sigma_{\mathrm{q}}$ to σ_{q} and did the same approximation to a polynomial of order four and calculated its derivative. Figure 8 shows this cross check for $\frac{\partial L}{\partial x_i}$ and $\frac{\partial L}{\partial I_i}$. Although the derivatives are similar, they are not exactly the same for $\frac{\partial L}{\partial x_i}$. This is caused by the polynomial coarseness fit, as explained below.

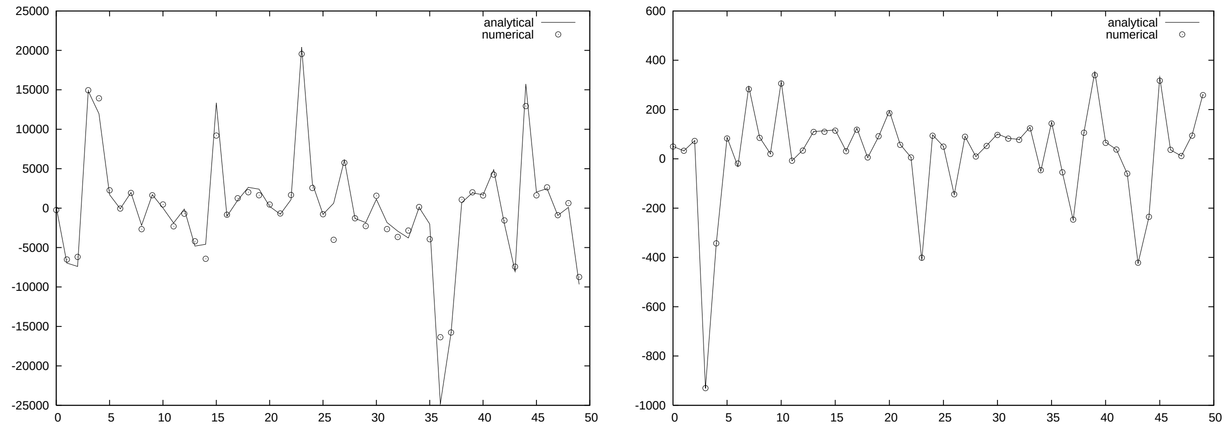

Fig. 8.— Verification of the derivatives. The solid line shows analytical derivatives, and dots show numerical approximations. *Left:* $\frac{\partial L}{\partial x_i}$. *Right:* $\frac{\partial L}{\partial I_i}$. The polygon identifier i is indicated on the x-axis.

Figure 9 shows the curve fit for $\frac{\partial L}{\partial x_i}$ for three different generators (generator number 37, 36 and 18 respectively). It can be seen in Figure 9 that the polynomial fit adjusts quite well to the function values for polygon number 37, so on Figure 8 both derivatives are the same. On the contrary, for polygons number 36 and 18, the fitted polynomial does not resemble the function L at δ=0, causing a slight difference in their derivatives on Figure 8. For polygon number 18 the polynomial does not fit the curve at all. This is the main problem of using a numerical approximation for the derivatives of $\{\vec{x_i}\}$: when two polygons are closer than δ, the generator displacement causes the function L to change abruptly (see Figure 10). [h]

It can be seen that the analytical and numerical derivatives on Figure 8 are almost the same. As explained above, differences are produced because there are cases where the polynomials do not fit well to the variations of the merit function L (for example, when two generators are too close). In an accurate calculation it is necessary to use the analytical derivatives.

B. Fitting a Voronoi Tessellation to an Image

Once we have a reasonable reconstruction for a pixelated image, we would like to fit a Voronoi tessellation to it in order to have a good initial starting point for the CG. This is done in an incremental way.

We start with a mesh consisting in only one polygon. We calculate the error per polygon as

$$e_i^2 = \sum_l (I_i - I_l^{\mathrm{im}})^2, \qquad \text{(B1)}$$

where the sum runs over all the pixels that fall inside polygon i, I_i is the intensity of that polygon and I_l^{im} is the intensity of pixel l in the image to be fitted. In each iteration we add a new polygon inside the one with the greatest error. The new generator is inserted in the position of the pixel that has the most different intensity value with respect to the mesh intensity.

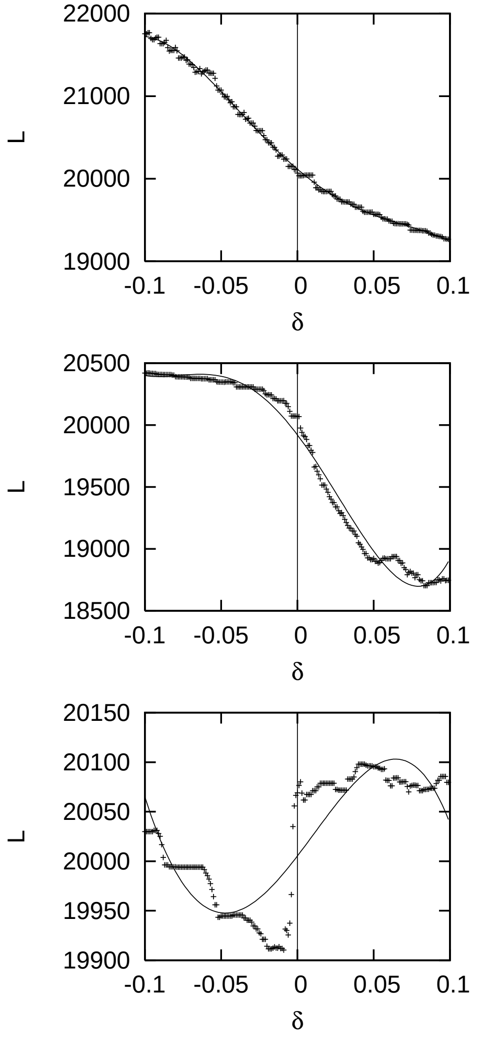

Fig. 9.— Examples of polynomial fits, used to determine numerical derivatives of the optimization function L for a particular generator. Dots represent L vs the polygon displacement δ in x and the solid line shows the fourth order polynomial fit to L. A vertical line is drawn at δ=0, where the derivatives were calculated. *Top:* Generator number 37, with a satisfactory polynomial fit. *Middle:* Generator number 36, the curve is not a good fit at δ=0. *Bottom:* 18th generator, the curve is not a good fit because L shows an abrupt variation near δ=0, which is due to the proximity of another generator.

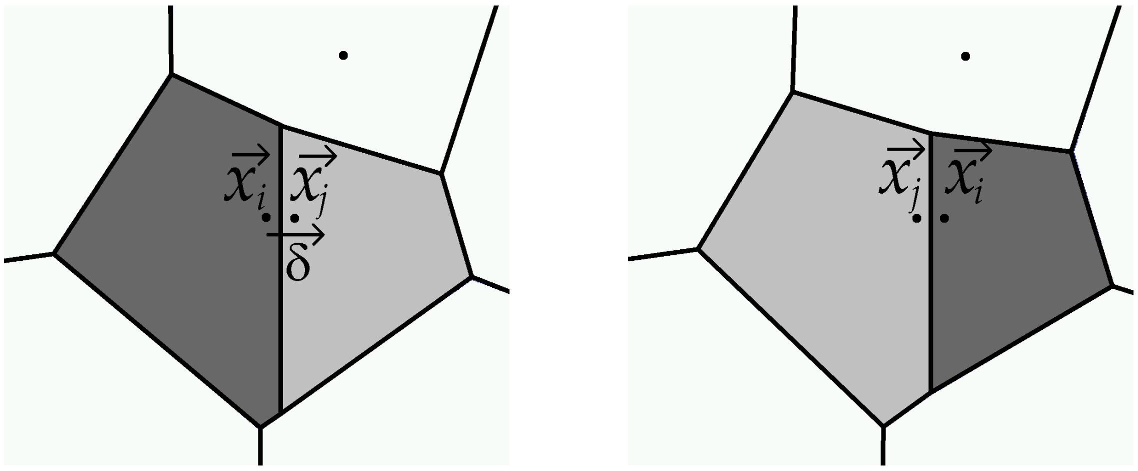

Fig. 10.— Translation of a generator close to another. *Left:* Before moving generator $\vec{x_i}$, polygon i, the darker polygon in the image, is on the left. *Right:* After moving generator $\vec{x_i}$, by a displacement of δ, polygon i is on the right of polygon j. When displacing generator $\vec{x_i}$ the diagram changes considerably, with a concomitant abrupt variation in L.

REFERENCES

Briggs, D. S., Schwab, F. R. & Sramek R. A. 1999, ASP Conf. Ser., 180, 127

Casassus, S., Cabrera, G. F., Förster, F, Pearson, T. J., Readhead, A. C. S., Dickinson, C. 2006, ApJ, 639, 951

Cornwell, T. J. & Evans, K. F. 1985, A&A, 143, 77

Högbom, J. A. 1974, A&AS, 15, 417

Narayan, Ramesh & Nityananda, Rajaram 1986, ARA&A, 24, 127

Okabe, A., Boots, B. & Sugihara, K. 1992, Spacial Tessellations Concepts and Applications of Voronoi Diagrams, John Wiley & Sons

Padin, S., et al, 2002, PASP, 114, 83

Piña, R. K. & Puetter, R. C. 1993, PASP, 105, 630

Press, W. H., Flannery, B. P., Teukilsky, S. A., Vettering, W. Y. 1992, Numerical Recipes in C, C. Cambridge University Press

Shepherd, M.C., 1997, in ASP Conf. Ser. 125, Astronomical Data Analysis Software and Systems VI, ed. G. Hunt & H.E. Payne (San Francisco: ASP), 77

Sutton, E. C. & Wandelt, B. D. 2006, ApJS, 162, 401

This 2-column preprint was prepared with the AAS LaTeX macros v5.2.