



Figure 1: Conversion from polar intensities $I_\nu(r, \theta)$ to a regular cartesian grid $I'_\nu(x, y)$. The r_i and θ_j gridpoints are drawn as black lines, the orange box represents a regular image pixel between cartesian grid points x_n and y_k . The dashed box shows the construction of a rectangle that exactly contains the polar cell (i, j) .

For Rosina

Peter Woitke, June 2013

1 A flux-conserving scheme to convert images in polar coordinates to regular cartesian grids

Coninuum images and channel maps in ProDiMo are initially calculated on a polar grid (see Thi et al. 2011 for details) with roughly¹ logarithmic equidistant radial grid points $\{r_i | i \in [0 \dots N_r]\}$ and linear equidistant angular grid points $\{\theta_j | j \in [0 \dots N_\theta]\}$. This is necessary to resolve the tiny inner disk rim (important for the short wavelengths) as well as resolving the outer regions (important for the long wavelengths) at the same time. The polar intensities $I_\nu(i, j)$ have associated areas in the image plane as

$$A_{ij} = \pi(r_i^2 - r_{i-1}^2)/N_\theta \quad \text{for } i \in [1 \dots N_r], j \in [1 \dots N_\theta] \quad (1)$$

The intensities are assumed to be constant on the polar “pixels”, i. e. the area A_{ij} bracketed by r_i, r_{i-1}, θ_j and θ_{j-1} , see Fig. 1.

We seek a fast numerical method to convert these polar images onto a regular grid with equidistant cartesian image coordinates $\{x_n | n \in [0 \dots N_x]\}$ and $\{y_k | k \in [0 \dots N_y]\}$ with associated pixel areas

$$A_{nk} = (x_n - x_{n-1})(y_k - y_{k-1}) \quad \text{for } n \in [1 \dots N_x], k \in [1 \dots N_y] \quad (2)$$

The method is flux-conservative if

$$\sum_{i=1}^{N_r} \sum_{j=1}^{N_\theta} A_{ij} I_\nu(i, j) = \sum_{n=1}^{N_x} \sum_{k=1}^{N_y} A_{nk} I'_\nu(n, k) \quad (3)$$

¹Some adjustments inside the inner rim and increased resolution towards the outer disk radius and beyond.

where $I'_\nu(n, k)$ are the desired intensities on the regular cartesian pixels. The exact solution of this problem would be to calculate the overlap areas, $O(A_{nk}, A_{ij})$, between the regular pixels A_{nk} and any polar pixel A_{ij} , then sum up the fluxes and divide by the pixel area as

$$I'_\nu(n, k) = \sum_{i=1}^{N_r} \sum_{j=1}^{N_\theta} \frac{O(A_{nk}, A_{ij})}{A_{nk}} I_\nu(i, j) , \quad (4)$$

but to calculate those overlap areas is painful, see Fig. 1. A more practical idea is to create a cartesian rectangular area that exactly contains the polar pixel, see Fig. 1, by taking the minimum and maximum of the four corner points of A_{ij}

$$\begin{aligned} x_l &= \min\{x_1, x_2, x_3, x_4\} & y_l &= \min\{y_1, y_2, y_3, y_4\} \\ x_r &= \max\{x_1, x_2, x_3, x_4\} & y_r &= \max\{y_1, y_2, y_3, y_4\} \\ x_1 &= r_{i-1} \sin(\theta_{j-1}) & y_1 &= r_{i-1} \cos(\theta_{j-1}) \\ x_2 &= r_{i-1} \sin(\theta_j) & y_2 &= r_{i-1} \cos(\theta_j) \\ x_3 &= r_i \sin(\theta_{j-1}) & y_3 &= r_i \cos(\theta_{j-1}) \\ x_4 &= r_i \sin(\theta_j) & y_4 &= r_i \cos(\theta_j) \end{aligned} \quad (5)$$

The area of this rectangular pixel,

$$A'_{ij} = (x_r - x_l)(y_r - y_l) , \quad (6)$$

is always larger than the area of the original polar pixel A_{ij} , resulting in a correction factor in Eq. (7) below. The area overlaps between A_{nk} and A'_{ij} are now easy to calculate, and the resulting conversion formula is

$$\begin{aligned} C_{nk} &= \sum_{i=1}^{N_r} \sum_{j=1}^{N_\theta} O(A_{nk}, A'_{ij}) \frac{A_{ij}}{A'_{ij}} \\ I'_\nu(n, k) &= \frac{1}{C_{nk}} \sum_{i=1}^{N_r} \sum_{j=1}^{N_\theta} O(A_{nk}, A'_{ij}) \frac{A_{ij}}{A'_{ij}} I_\nu(i, j) , \end{aligned} \quad (7)$$

The area coverage C_{nk} is close, but not exactly equal to A_{nk} , because some regular pixels will be slightly oversampled by the polar pixels, and others slightly undersampled, and this effect is automatically corrected for in Eq. (7).

This conversion method is fast, reliable, and exactly flux conservative. However, the resulting spatial resolution in the regular image will suffer somewhat in areas where the rectangular pixels A'_{ij} , as constructed from the polar pixels A_{ij} , are larger than the regular pixels A_{nk} . Here, the simulation can be improved by introducing polar sub-pixels in the first place, i.e. by subdividing large polar pixels into a suitable number of smaller polar sub-pixels, equally spaced in r and θ , assuming the intensities to be constant over all sub-pixels, before applying Eqs. (5) to (7).

References

Thi, W.-F., Woitke, P., & Kamp, I. 2011, MNRAS, 412, 711