# Spatial Characterization of Tray-Dependent Beam Structure and Exploratory Corner Geometry in a Preclinical Irradiator

Radiation and Environmental Biophysics

Bashayr Alotaibi[1], Will Ren[2], Bonny Dikago[1], Ara Alexandrian[3], Joseph Steiner[4], and Remo George[1]*

[1] Nuclear Medicine and Molecular Imaging Sciences Program, Department of Clinical and Diagnostic Sciences, School of Health Professions, University of Alabama at Birmingham, Birmingham, AL, USA

[2] Hoover High School, Hoover, AL, USA

[3] Department of Medical Physics, Mary Bird Perkins Cancer Center, Baton Rouge, LA, USA

[4] Department of Radiology, University of Chicago, Chicago, USA

*Corresponding author:

Remo George, Ph.D.
Nuclear Medicine and Molecular Imaging Sciences Program,
Department of Clinical and Diagnostic Sciences
School of Health Professions
University of Alabama at Birmingham
Birmingham, AL, USA
Email: remogeorge@uab.edu
ORCID: https://orcid.org/0000-0003-3108-2056

Online Resource 1

**Upper Level**

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D

# Load the Excel file
file_path = 'Beam Profile UT.xlsx'
df = pd.read_excel(file_path, index_col=0)

# Convert the data to NumPy arrays
x = df.columns.astype(float)  # X-coordinates from columns
y = df.index.astype(float)    # Y-coordinates from index
X, Y = np.meshgrid(x, y)
Z = df.values                 # Dose values

# Plotting the 3D surface
fig = plt.figure(figsize=(10, 7))
ax = fig.add_subplot(111, projection='3d')

surf = ax.plot_surface(X, Y, Z, cmap='viridis',
                       edgecolor='none')
fig.colorbar(surf, ax=ax, shrink=0.5, aspect=5)

# Labels and title
ax.set_xlabel('X (cm)')
ax.set_ylabel('Y (cm)')
ax.set_zlabel('Dose (Gy)')
ax.set_title('3D Dose Profile (Upper Tray)')

plt.tight_layout()
plt.show(block=True)
```

**Lower Level**

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D

# Load the Excel file
file_path = 'Beam Profile LT.xlsx'  # Ensure this file is in the same folder!
sheet_name = 'Sheet1'  # Update if your sheet has a different name

# Read data
df = pd.read_excel(file_path, sheet_name=sheet_name, header=None)

# Extract Y-axis positions (first column, excluding last row)
y_coords = df.iloc[:-1, 0].str.replace('cm', '').astype(float).values

# Extract X-axis positions (last row, excluding first column)
x_coords = df.iloc[-1, 1:].str.replace('cm', '').astype(float).values

# Extract dose values (all rows except last, all columns except first)
dose_values = df.iloc[:-1, 1:].replace('', 0).astype(float).values

# Add these checks:
print("X positions (cm):", x_coords)
print("Y positions (cm):", y_coords)
print("Dose values shape:", dose_values.shape)
print("Max dose value:", np.max(dose_values))

# Create a grid for 3D plotting
X, Y = np.meshgrid(x_coords, y_coords)

# Create the 3D plot
fig = plt.figure(figsize=(12, 8))
ax = fig.add_subplot(111, projection='3d')

# Surface plot
surf = ax.plot_surface(
    X, Y, dose_values,
    cmap='viridis',
    edgecolor='none',
    rstride=1,
    cstride=1
)

# Add color bar
fig.colorbar(surf, ax=ax, label='Dose (Gy)')

# Labels and title
ax.set_xlabel('X Position (cm)')
ax.set_ylabel('Y Position (cm)')
ax.set_zlabel('Radiation Dose')
ax.set_title('3D Radiation Dose Profile in Biological Cabinet')

# Adjust view angle
ax.view_init(elev=30, azim=45)

# Show the plot
plt.tight_layout()
plt.show()
```

**Online Resource 1 Python scripts used to generate the 3D dose surface plots shown in figure 2(a, b).** The scripts imported coordinate and dose data, constructed mesh grids, and rendered oblique, lateral, and top surface views using matplotlib. Axis labels and visualization parameters corresponded to those used in the final figures

Online Resource 2

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from matplotlib.colors import LinearSegmentedColormap

# Load data
df = pd.read_excel("Beam_Profile_LT_Clean_Copy.xlsx.xlsx", index_col=0)
Z = df.values
x = df.columns.astype(float)
y = df.index.astype(float)
X, Y = np.meshgrid(x, y)

# Calculate levels from 10% to 100% in 10% increments
max_dose = Z.max()
levels = [round(i*max_dose, 4) for i in np.arange(0.1, 1.1, 0.1)]  # 10% to 100%

# Medical-grade color map (cool to warm)
colors = [
    '#9900FF',  # Purple (10%)
    '#0066FF',  # Blue (20%)
    '#0099FF',  # Light Blue (30%)
    '#00CC00',  # Green (40%)
    '#66FF00',  # Lime (50%)
    '#FFCC00',  # Yellow (60%)
    '#FF6600',  # Orange (70%)
    '#FF3300',  # Red-Orange (80%)
    '#FF0000',  # Red (90%)
    '#CC0000'   # Dark Red (100%)
]

# Create figure with medical physics standards
plt.figure(figsize=(10, 8))
plt.gca().set_aspect('equal')

# Filled contours for color reference
cf = plt.contourf(X, Y, Z, levels=levels, colors=colors, alpha=0.25)

# Contour lines
contours = plt.contour(X, Y, Z, levels=levels, colors='k', linewidths=1.2)

# Color bar with percentage labels
cbar = plt.colorbar(cf, ticks=levels, pad=0.02)
cbar.set_label('Percentage of Max Dose', rotation=270, labelpad=25)
cbar.set_ticklabels([f'{i*100:.0f}%' for i in np.arange(0.1, 1.1, 0.1)])

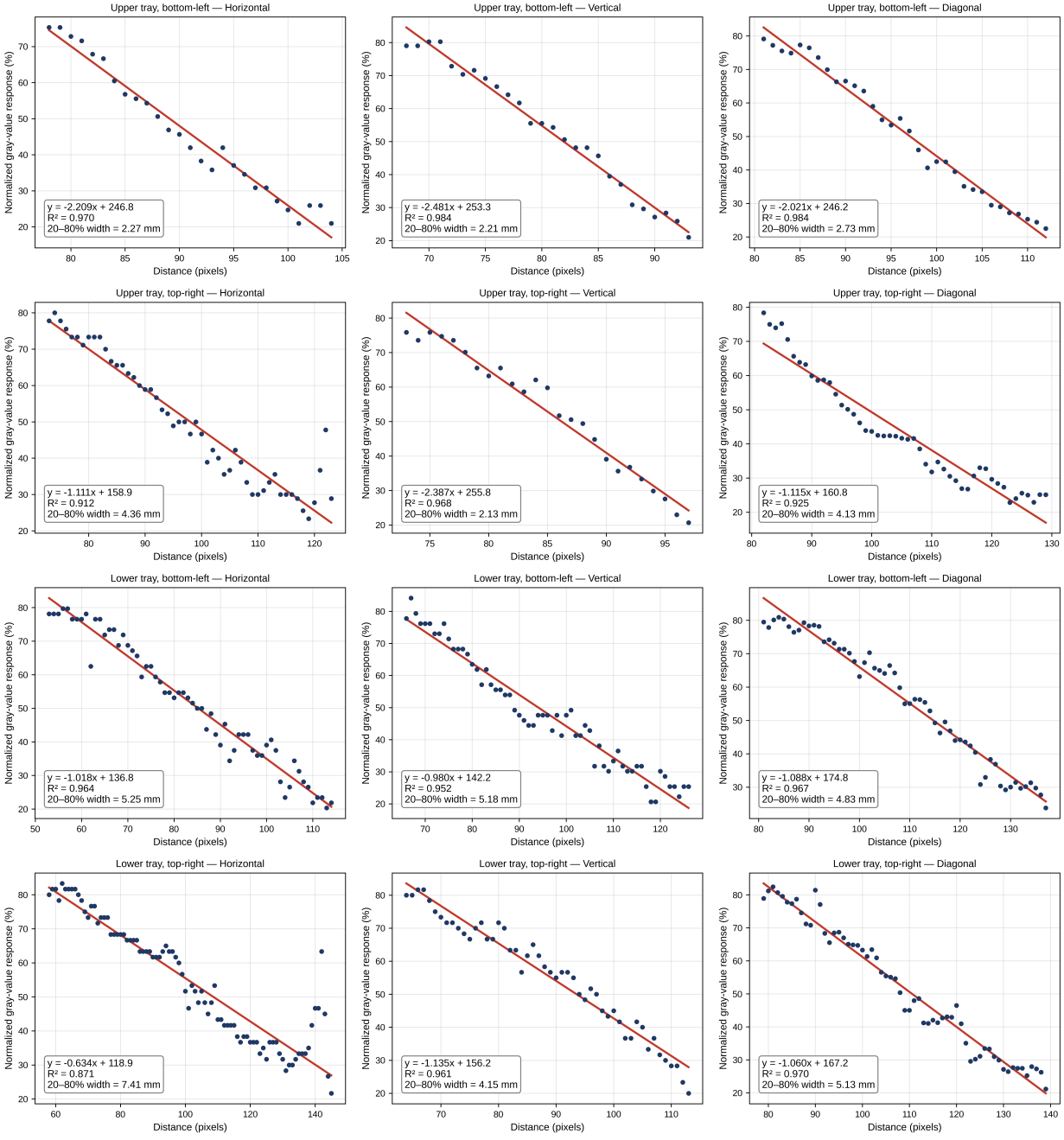
# Formatting
plt.xlabel("X Position (cm)", fontsize=12)
plt.ylabel("Y Position (cm)", fontsize=12)
plt.title("Isodose Contour Plot (Lower Tray)", fontsize=14, pad=20)
plt.gca().invert_yaxis()  # Medical convention
plt.grid(True, linestyle=':', alpha=0.3)

# Ensure 100% level is visible (adjust if needed)
if max_dose in Z:
    plt.scatter(X[Z == max_dose], Y[Z == max_dose], color='#CC0000',
                marker='*', s=100, label='100% Isodose')
    plt.legend()

plt.tight_layout()
plt.savefig("Isodose_LT_FullRange.png", dpi=300, bbox_inches='tight')
plt.show()
```

**Online Resource 2 Python script used to generate the isodose contour plots shown in figure 3.** The code processes gridded dose data, calculates percentage isodose levels, and produces filled and line contour visualizations with equal axis scaling. Formatting elements include labeled axes, a platform-specific title, and a marker identifying the 100% isodose location

Online Resource 3




**Online Resource 3 Supplementary Fig. S1. Local linear fits of the 20 - 80% normalized gray value transitions for the four evaluated corner films.** The panels show descriptive plots of the measured data points and ordinary least-squares regression lines for the horizontal, vertical, and diagonal profiles obtained at the Upper Bottom-Left, Upper Top-Right, Lower Bottom-Left, and Lower Top-Right positions. Each panel displays the regression equation, coefficient of determination ($R^2$), and 20 - 80% film response transition width. The horizontal axis values represent pixel positions, with physical distances calculated using a conversion factor of 0.0847 mm per pixel. Values displayed under the vertical axis label “Dose (%)” represent normalized film gray value. The abbreviation “Pen” within each panel refers to the 20 - 80% normalized gray value transition width in millimeters