**Milestone 1: Hierarchical Block Matching Code for Generating Disparity Maps**

**Job Description: 6 days**

Develop a Python-based code for generating disparity maps from stereo images using a hierarchical block matching algorithm. The code should work with any stereo image pair and include advanced functionalities for identifying unconfident matches, generating connected regions, and calculating bounding box disparities.

**Scope of Work:**

1. **Disparity Map Generation**
   * Use hierarchical block matching with the left image as the reference.
   * Begin with a block size of 32x32. If no match is found, split the block into four 16x16 blocks, then further into 8x8 blocks as necessary.
   * Perform an exhaustive search around blocks if no match is found using the following criteria:
     + Right: Stop at 4 pixels.
     + Left: Stop at the maximum disparity value.
     + Other directions: Stop at 2 pixels.
   * Use Mean Squared Error (MSE) for block comparison, starting with a high initial value and updating the minimum MSE dynamically.
   * Normalize the disparity map to values between 0-255.
2. **Texture Sensitivity**
   * Adjust acceptable MSE thresholds based on texture levels using standard deviation:
     + High texture: Accept higher MSE thresholds.
     + Low texture: Use lower MSE thresholds (for areas like sky or dark objects).
3. **Uncertainty and Confidence Mapping**
   * Identify unconfident matches during block matching and generate uncertainty maps for both images.
   * Use the Marching Squares Algorithm to connect unconfident blocks into continuous contours.
4. **Bounding Box Analysis**
   * Fit bounding boxes around the contours and extract properties:
     + Position (x, y), width, height, area, and average color.
   * Match bounding boxes between the left and right images using:
     + Position difference (Euclidean distance).
     + Size and area difference.
     + Color difference (Euclidean distance).
5. **Bounding Box Disparity Calculation**
   * Calculate disparity from the x-coordinate difference of bounding box centers.
   * Update the disparity map using bounding box disparities, ensuring accurate depth representation in textureless regions.

**Expected Outputs:**

* **Maps and Visualizations:**
  + Initial grayscale disparity map (must closely resemble a provided reference map).
  + HSV color disparity map converted from grayscale.
  + Binary confidence maps for both left and right images.
  + MSE histogram in HSV color format.
  + Marching Squares maps for both left and right images showing connected regions.
  + Bounding box maps for both left and right images with visible bounding boxes.
  + Updated grayscale and color disparity maps.
* **Data Files:**
  + CSV file of block matching details including positions, MSE, disparity, standard deviation, and search directions.
  + CSV file of bounding box details including properties and matching results.

**Requirements:**

* The code must:
  + Work with any stereo image.
  + Generate outputs suitable for provided test cases, including a CG stereo image and a real-life stereo image (with features like clouds in the sky).
  + Draw shapes around challenging areas (e.g., clouds) in the final test image.

**Deliverables:**

* Complete Python code with comments.
* Instructions for running the code with any stereo image.

**Additional Notes:**

* A reference disparity maps will be provided for comparison.