**1. Setup and Data Processing:**

* **Environment Setup:**
  + Install and configure necessary Python libraries (numpy, matplotlib, svgwrite, cairosvg).
* **Data Handling:**
  + Implement functions to read CSV files and convert them to the required data structures.
  + Create visualization functions using matplotlib and SVG conversion.

**2. Curve Regularization:**

* **Algorithm Implementation:**
  + Detect and regularize various geometric shapes:
    - Straight lines
    - Circles and ellipses
    - Rectangles and rounded rectangles
    - Regular polygons
    - Star shapes
* **Testing:**
  + Use isolated.csv and fragmented examples (frag0.csv, frag1.csv, frag2.csv) for testing.

**3. Symmetry Detection:**

* **Detection Implementation:**
  + Implement reflection symmetry detection for closed shapes.
  + Apply symmetry detection to regularized curves.
* **Testing:**
  + Test with the same set of examples as in curve regularization.

**4. Curve Completion:**

* **Algorithm Implementation:**
  + Implement algorithms for completing curves with occlusions:
    - Connected occlusion (occlusion1.csv)
    - Disconnected occlusion (occlusion2.csv)
* **Completion Requirements:**
  + Ensure smooth, regular, and symmetric completions where appropriate.

**5. Integration and Refinement:**

* **Pipeline Integration:**
  + Combine all components into a cohesive pipeline.
  + Implement the polylines2svg function for output generation.
* **Algorithm Refinement:**
  + Refine algorithms based on test results.

**6. Testing and Validation:**

* **Test Suite Creation:**
  + Develop a comprehensive test suite using provided examples.
* **Result Validation:**
  + Validate results against provided solutions (e.g., isolated\_sol.csv, frag01\_sol.csv).
* **Rasterization:**
  + Use rasterization for evaluating occlusion solutions.

**7. Optimization and Documentation:**

* **Algorithm Optimization:**
  + Optimize algorithms for performance.
* **Documentation:**
  + Thoroughly document code, explaining approaches and assumptions.