Jigsaw Image Mosaics (JIM)

Based on the paper ‘Jigsaw Image Mosaics’ by Junhwan Kim and Fabio Pellacini, SIGGRAPH 2002

April 1, 2004

Presentation by Kaleigh Smith

Outline

- Description of JIM, artistic inspiration and related work
- Definition of JIM Energy Framework
- Input Preparation and Active Contours
- Mosaic Algorithm
- Algorithm Optimizations
- JIM Results
- Comments on JIM

JIM: Jigsaw Image Mosaic

Junhwan Kim
Fabio Pellacini

SIGGRAPH
2002

Arcimboldo
1527-1593

Summer,
1573

Photomosaic

Robert Silvers and Michael Hawley

1997

Photomosaic

- Fixed container shape (rectangular)
- Fixed tile shape (rectangular)
- Fixed packing (grid)

Photomosaic

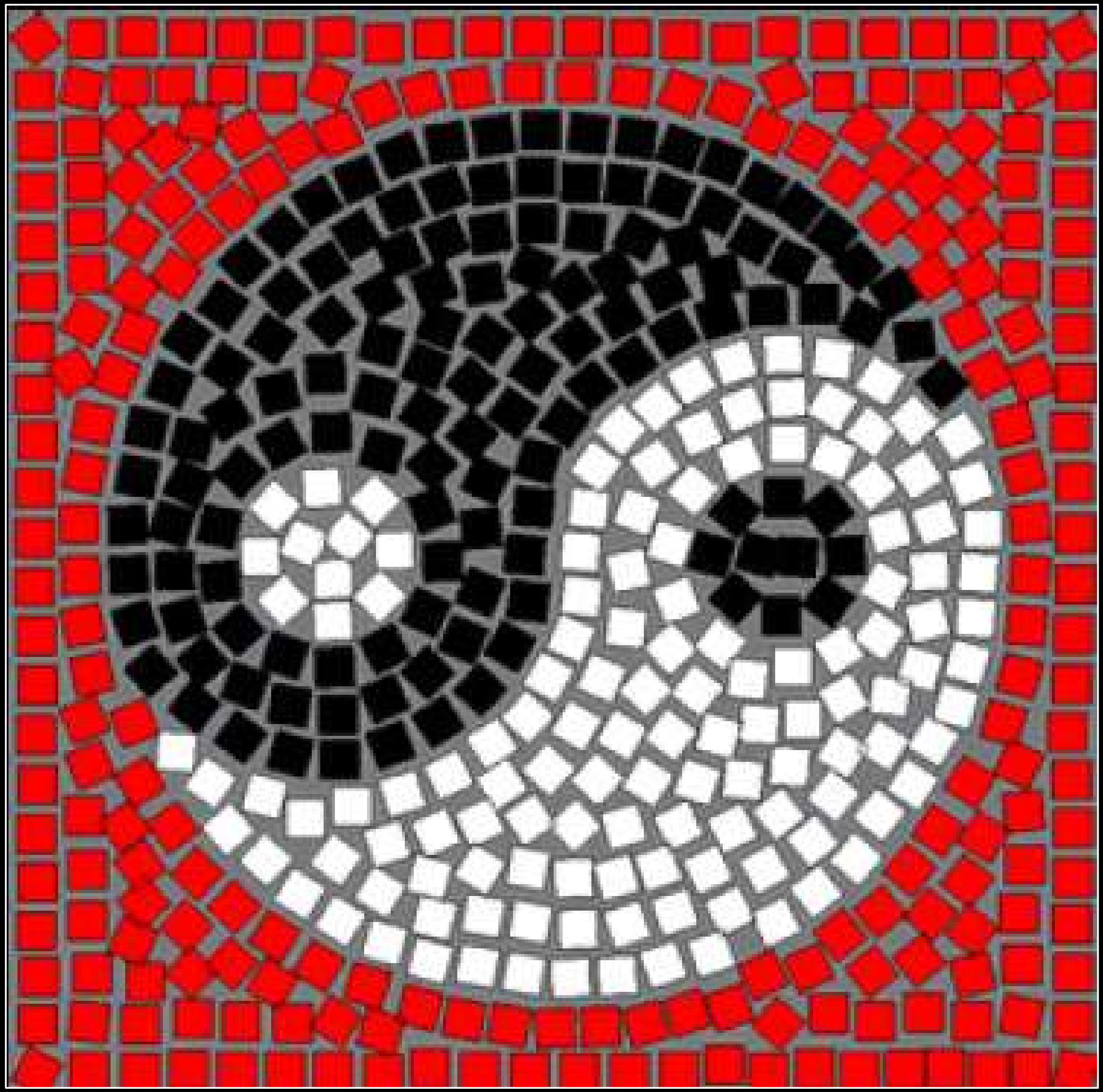
- Fixed container shape (rectangular)
- Fixed tile shape (rectangular)
- Fixed packing (grid)

- Match the intensity of the tile texture to the underlying image intensity. No special packing.

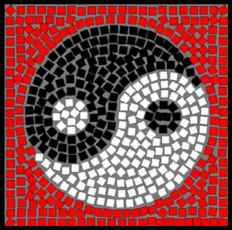
Simulated Decorative Mosaic

Alejo Hausner

SIGGRAPH 2001

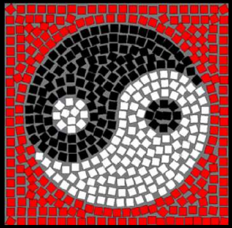


Simulated Decorative Mosaic



- Fixed container shape (rectangular)
- Fixed tile shape (rectangular or elliptical)
- Important image edges represented by user-specified feature curves.

Simulated Decorative Mosaic



- Fixed container shape (rectangular)
- Fixed tile shape (rectangular or elliptical)
- Important image edges represented by user-specified feature curves.

- Determines best packing of tiles in container and orients tiles to feature curves to preserve edges from the source image.
- Allows tile configuration to have gaps and overlapped tiles.

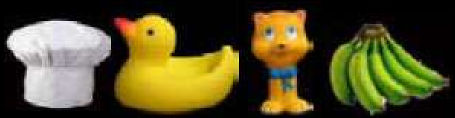
Outline

- Description of JIM, artistic inspiration and related work
- Definition of JIM Energy Framework
- Input Preparation and Active Contours
- Mosaic Algorithm
- Algorithm Optimizations
- JIM Results
- Comments on JIM

JIM: Jigsaw Image Mosaic

- Properties of a JIM - arbitrarily shaped container, arbitrarily shaped tiles of textures.

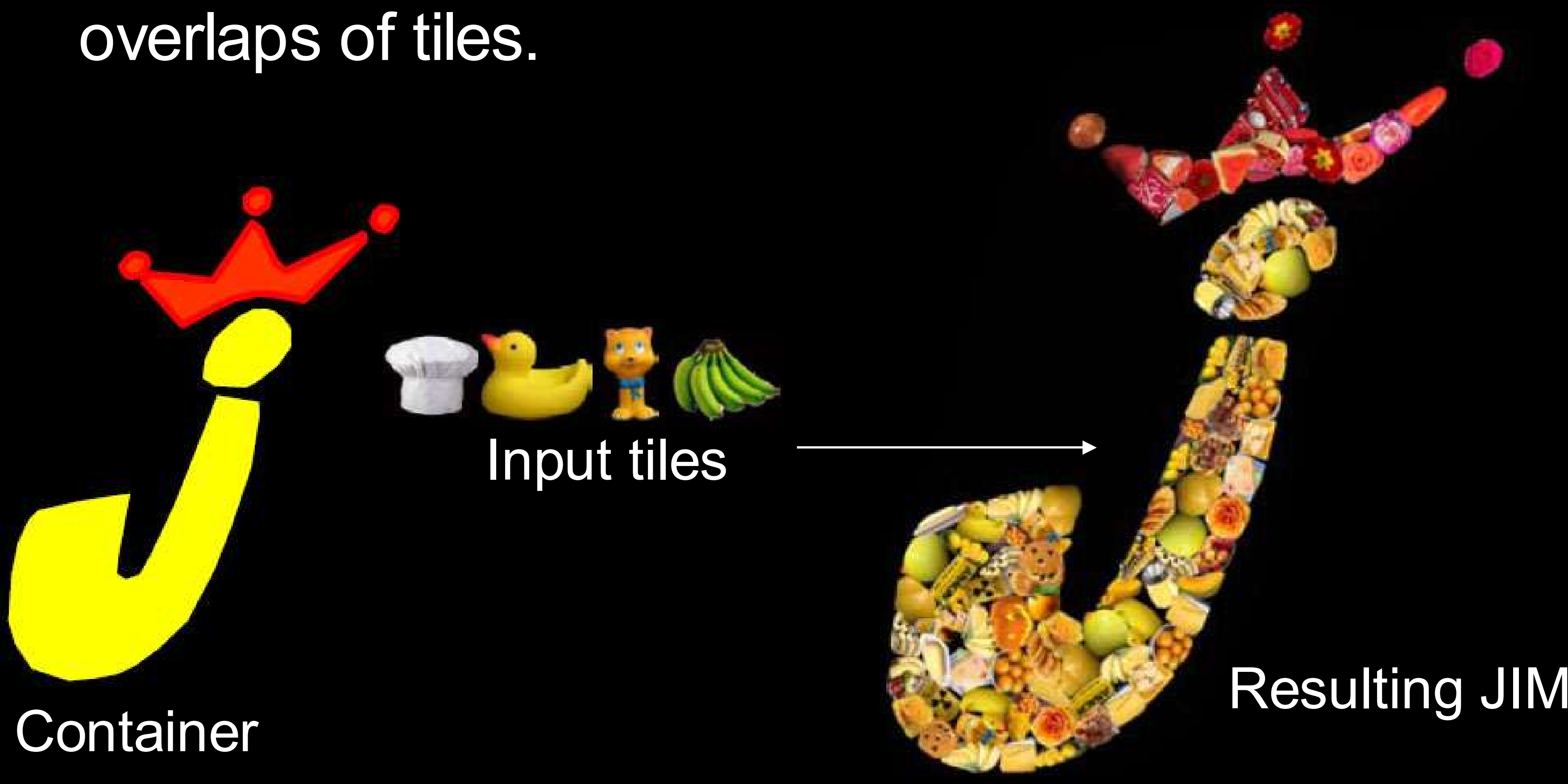
Container



Input tiles

JIM: Jigsaw Image Mosaic

- Properties of a JIM - arbitrarily shaped container, arbitrarily shaped tiles of textures.
- Tiles packed arbitrarily and allows for gaps and overlaps of tiles.



JIM: Jigsaw Image Mosaic

- JIM approaches problem as an energy minimization problem, where the energy of a mosaic is a sum of mosaic-related energy terms.

JIM: Jigsaw Image Mosaic

- JIM approaches problem as an energy minimization problem, where the energy of a mosaic is a sum of mosaic-related energy terms.

- Claim that JIM generalizes mosaics by creating a generalized framework.
  - "Energy-based framework for the mosaicing problem which generalizes on known algorithms"

- Question: is this claim true or proven true by the paper?

JIM: Jigsaw Image Mosaic

- **Tile Configuration:** subset of input tiles with repetition, along with their associated transformations (orientation, translation, deformation).

JIM: Jigsaw Image Mosaic

- **Tile Configuration:** subset of input tiles of the input tiles with repetition, along with their associated transformations.

- **JIM:** a tile configuration that minimizes energy E.

$$E = W_C E_C + W_G E_G + W_O E_O + W_D E_D$$

colour gap overlap deformation

JIM: Energy Framework

- **Tile Configuration:** subset of input tiles of the input tiles with repetition, along with their associated transformations.

- **JIM:** a tile configuration that minimizes energy E.

$$E = W_C E_C + W_G E_G + W_O E_O + W_D E_D$$

colour gap overlap deformation

• How to produce photomosaic or decorative mosaic?

JIM: Energy Framework

$$E = W_C E_C + W_G E_G + W_O E_O + W_D E_D$$

colour gap overlap deformation

- The energy of a tile configuration is the sum of each weighted energy term.
- Each term is the sum of the energy term measured for each tile in the configuration.

JIM: Energy Framework

$$E = W_C\, E_C + W_G\, E_G + W_O\, E_O + W_D\, E_D$$

colour **gap** **overlap** **deformation**

- Terms can be added or removed (flexible and scalable framework).
- Terms can be measured with different metrics.

JIM: Energy Terms Evaluation

$$E = W_C E_C + W_G E_G + W_O E_O + W_D E_D$$

colour gap overlap deformation

- Colour: random locations on each tile, L2 differences.

JIM: Energy Terms Evaluation

$$E = W_C E_C + W_G E_G + W_O E_O + W_D E_D$$

colour gap overlap deformation

- Colour: random locations on each tile, L2 differences.
- Gap and Overlap: “spring energy formulation”.
  - Use the boundary shapes of the tiles and the container to determine the signed distance between each tile and the nearest tile or container edge.

JIM: Energy Terms Evaluation

$$E = W_C E_C + W_G E_G + W_O E_O + W_D E_D$$

colour gap overlap deformation

- Colour: random locations on each tile, L2 differences.
- Gap and Overlap: "spring energy formulation".
  - Use the boundary shapes of the tiles and the container to determine the signed distance between each tile and the nearest tile or container edge.
- Deformation: difference between original tile shape and deformed tile shape.

Outline

- Description of JIM, artistic inspiration and related work
- Definition of JIM Energy Framework
- Input Preparation and Active Contours
- Mosaic Algorithm
- Algorithm Optimizations
- JIM Results
- Comments on JIM

JIM: Preparation of Input

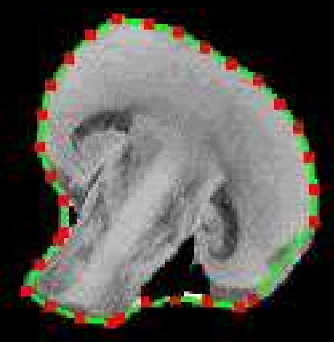
- JIM works on arbitrarily shaped containers and tiles.
  - The container and tile shapes are determined and represented using Active Contours.

- Also, active contours are used to segment a source image into a set of arbitrarily shaped containers.

JIM: Shapes by Active Contours

- Active Contours are a classic shape model described by Kass, Witkin and Terzopoulos, ‘Snakes: Active Contour Models’ (1987).

- Contour = vertices (control points) connected by edges.



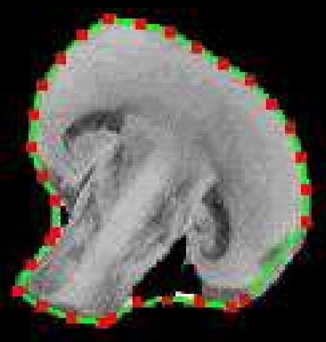
Source: Philip Lau and Katia Hristo

JIM: Shapes by Active Contours

- Contour is controlled by minimizing an energy function of properties: snake continuity, snake curvature and image gradient.

- We use them to find image boundary.

- Also used to deform image boundaries.



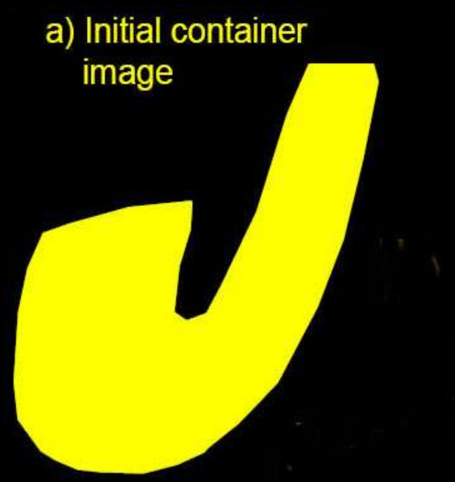
Source: Philip Lau and Katia Hristo

Outline

- Description of JIM, artistic inspiration and related work
- Definition of JIM Energy Framework
- Input Preparation and Active Contours
- Mosaic Algorithm
- Algorithm Optimizations
- JIM Results
- Comments on JIM

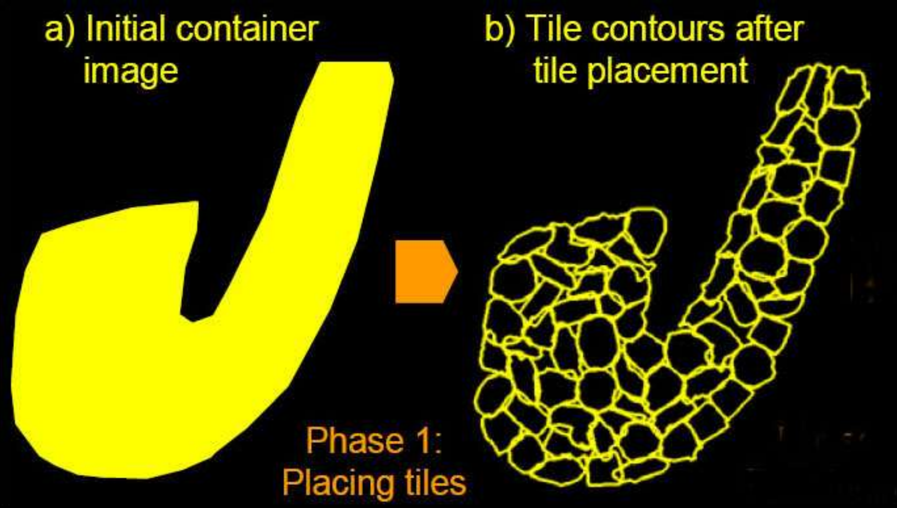
JIM: Mosaic Algorithm

- 1. Prepare input tiles, segment source image and treat each container separately.



JIM: Mosaic Algorithm

- 1. Prepare input tiles, segment source image and treat each container separately.
- 2. Pack the container with tiles from tile set.



JIM: Mosaic Algorithm

- **Best first search** for creating the packing.
  - 1. Find a suitable position in container – this gives a container region.
  - 2. Search for tile to use and register tile to the determined container region.
  - 3. Subtract tile shape from the container to get new container shape to pack.

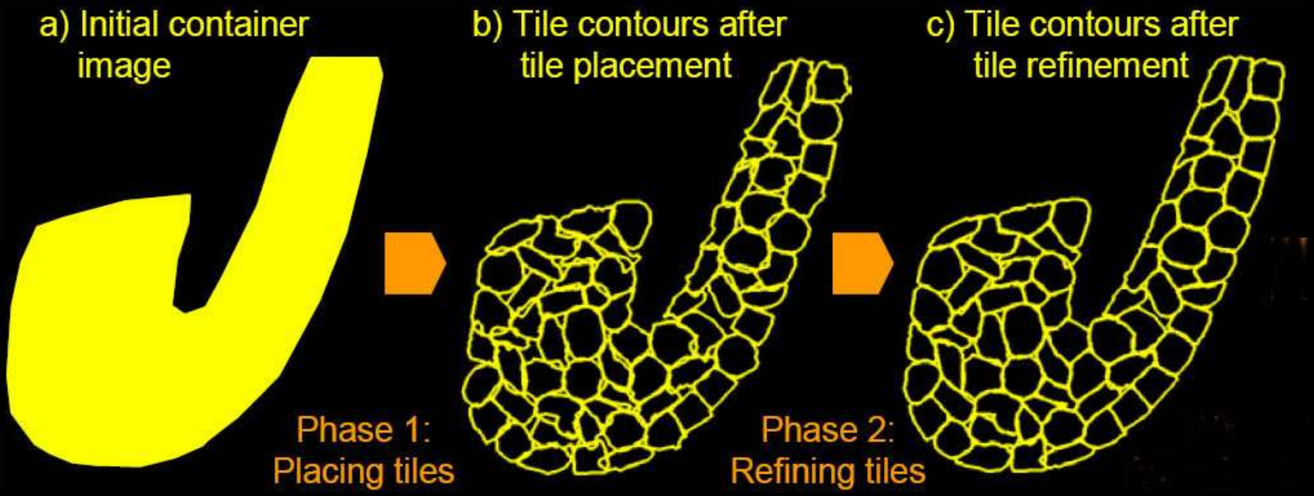
JIM: Mosaic Algorithm

- **Best first search** for creating the packing.
  - 1. Find a suitable position in container – this gives a container region.
  - 2. Search for tile to use and register tile to the determined container region.
  - 3. Subtract tile shape from the container to get new container shape to pack.

If can't find a tile to finish filling a container, backtrack to last configuration with minimal energy.

JIM: Mosaic Algorithm

- 1. Prepare input tiles, segment source image and treat each container separately.
- 2. Pack the container with tiles from tile set.
- 3. Refine the packing by deforming the tiles.



JIM: Mosaic Algorithm

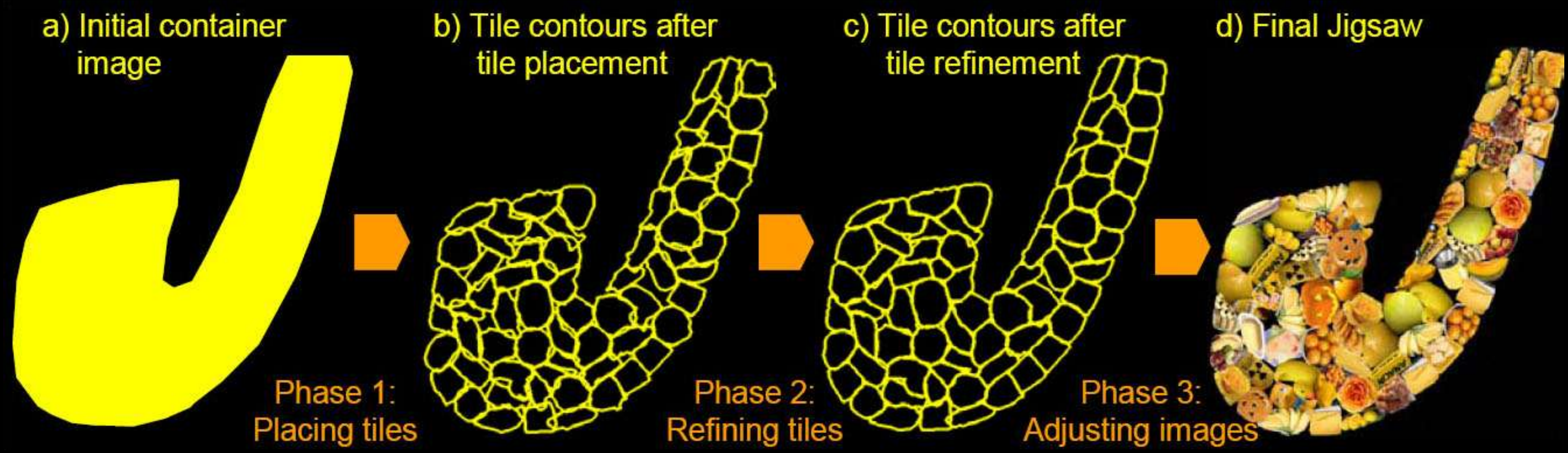
- **Refine the tile shapes.** Reduce gap or overlap.
  - Use a set of active contours and minimize energy according to forces that:
    - maintain contour original shape
    - repulse between two overlapping contours
    - attract two contours if they are separated by a gap.

JIM: Mosaic Algorithm

- **Refine the tile shapes.** Reduce gap or overlap.
  - Use a set of active contours and minimize energy according to forces that:
    - maintain contour original shape
    - repulse between two overlapping contours
    - attract two contours if they are separated by a gap.
- This minimizes over all four energy terms, and must not increase energy of a configuration.

JIM: Mosaic Algorithm

- 1. Prepare input tiles, segment source image and treat each container separately.
- 2. Pack the container with tiles from tile set.
- 3. Refine the packing by deforming the tiles.



JIM: Mosaic Algorithm

- The algorithm:

$$O(\,(V_{tile})(N_{tile})(V_{container})(N_{tilesInContainer})(1+b)\,)$$

- Number of vertices per tile.
- Number of tiles.
- Number of vertices per container.
- Number of tiles in the container.
- Branching overhead for backtracking in search.

Outline

- Description of JIM, artistic inspiration and related work
- Definition of JIM Energy Framework
- Input Preparation and Active Contours
- Mosaic Algorithm
- Algorithm Optimizations – i.e. GOOD IDEAS
- JIM Results
- Comments on JIM

JIM: Optimization 1

- REDUCE: Branching overhead for backtracking in search.
- Want to place tiles so that it is easy to fill container shape at each iteration of algorithm (no protrusions and container shape is convex).

JIM: Optimization 1

- REDUCE: Branching overhead for backtracking in search.
- Want to place tiles so that it is easy to fill container shape at each iteration of algorithm (no protrusions and container shape is convex).
  - Fill areas with least number of neighbours first. Use Centroidal Voronoi Diagram (CVD).
  - Add Lookahead energy term to energy formula that penalizes tiles that make container shape difficult to fill at next iteration.

JIM: Optimization 2

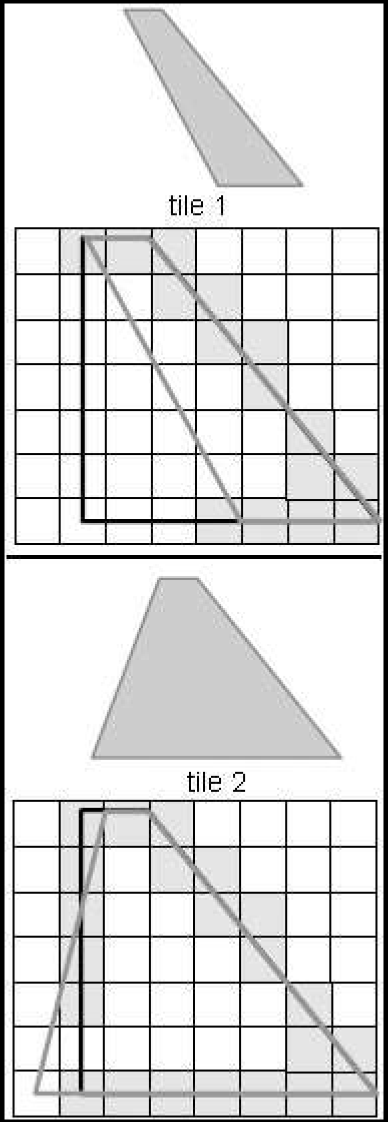
- REDUCE: Number of vertices representing container shape.
- At each iteration of the algorithm, the container shape changes due to the removal of the added tile.
- Results in jagged edges and container fragments.
    - If fragment is smaller than smallest tile, treat as a gap and remove from resulting container shape.

JIM: Optimization 3

- REDUCE: Number of tiles to be searched.
- At each iteration of the algorithm must search all tiles to find the tile which best fits into the container region to be filled (the predetermined best location to be filled).
- Use **Geometric Hashing** so that the algorithm does not consider tiles that are bad fits for the container region.

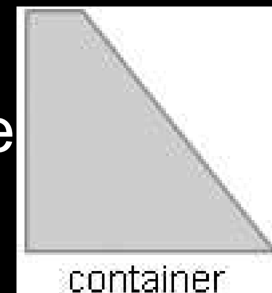
JIM: Optimization 3

- **Geometric Hashing** reduces number of tiles to search.
- Create grid of squares in plane. Each square corresponds to hash table entry.
- Place each tile and orientation over the grid and keep track of all tiles and their orientations that cross each square of the grid.

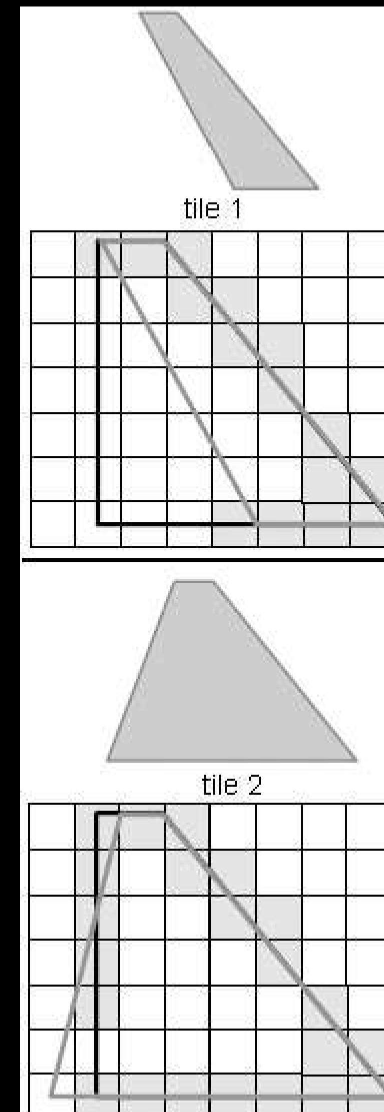


JIM: Optimization 3

- Take boundary of container region to be filled and align ove grid.



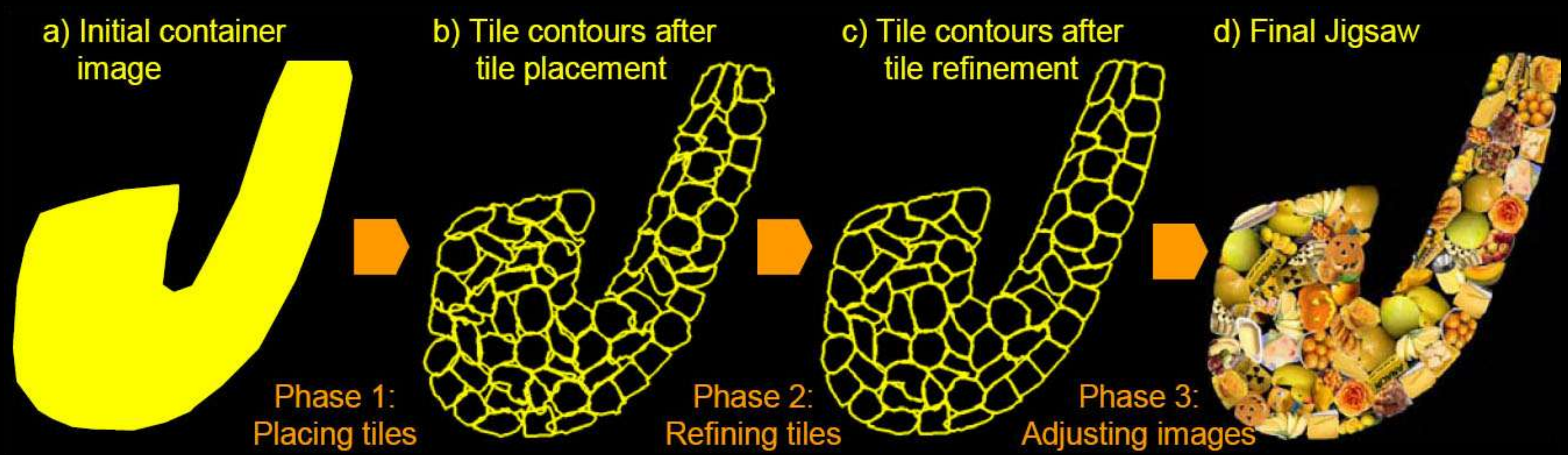
- For every grid square crossed by container region, have a list of all tiles and orientations that also crossed that square.



- Candidates for best fitting tile: the tiles that share the most crossed squares with the container region.

JIM: Mosaic Algorithm

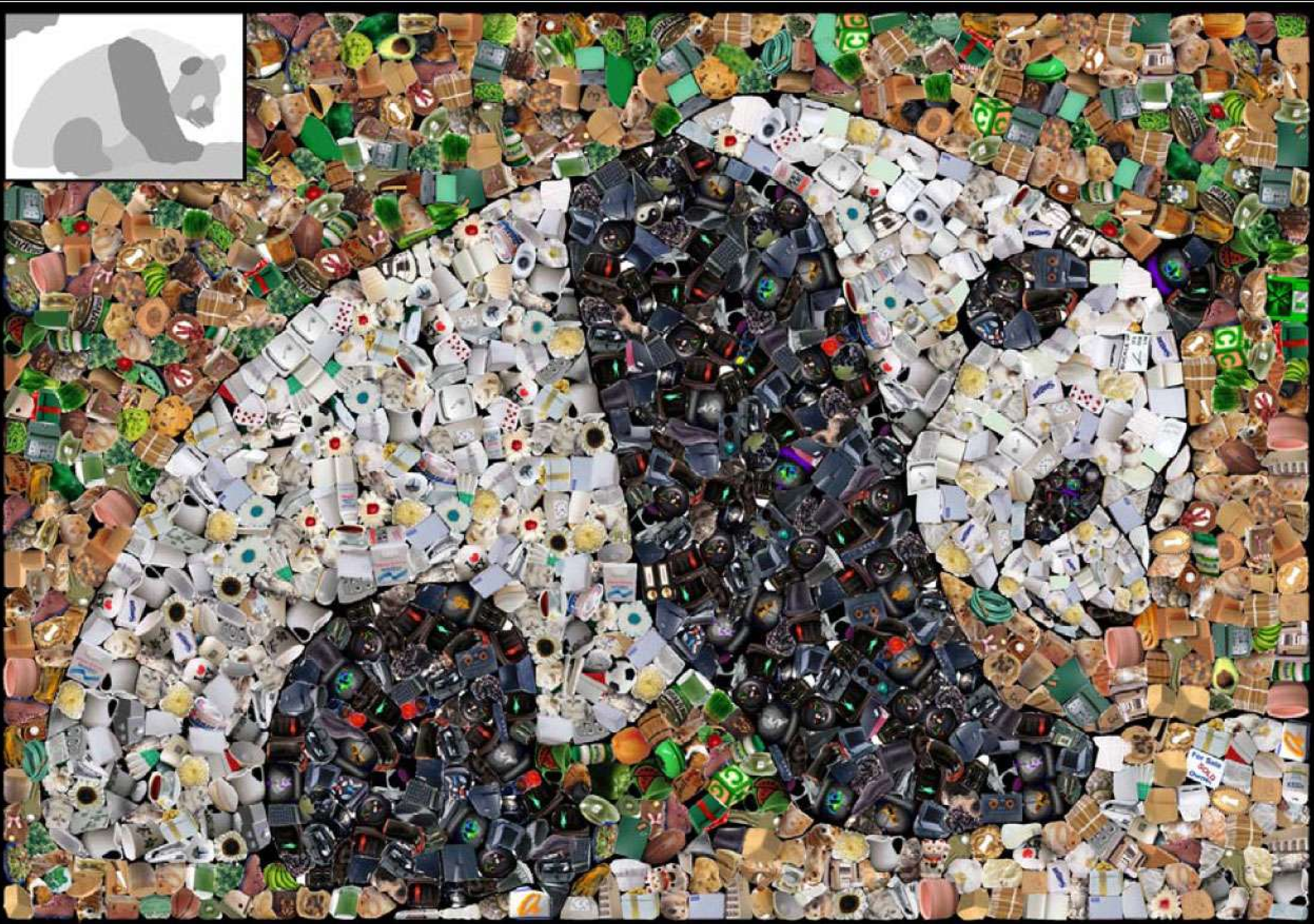
- So that's how they optimize the straightforward mosaic algorithm.



Outline

- Description of JIM, artistic inspiration and related work
- Definition of JIM Energy Framework
- Input Preparation and Active Contours
- Mosaic Algorithm
- Algorithm Optimizations
- JIM Results – acronym, panda and parrot
- Comments on JIM


jǐn



Outline

- Description of JIM, artistic inspiration and related work
- Definition of JIM Energy Framework
- Input Preparation and Active Contours
- Mosaic Algorithm
- Algorithm Optimizations
- JIM Results
- Comments on JIM

JIM: Comments

- "Energy-based framework for the mosaicing problem which generalizes on known algorithms"
    - Is this true?
    - There are no examples of a JIM that reproduces a simulated decorative mosaic.
    - Styles seem intuitively different, especially with respect to the tile orientation.
    - The framework has little to do with the actual physical process of creating a mosaic.

JIM: Comments

- “deforming them slightly to achieve a more visually-pleasing effect”
    - Again, subjective.
    - Does smooshing together really create a better mosaic?
    - Tile deformation increases the computer-created look of JIM but does not make it look more like a mosaic.

References

- “Jigsaw Image Mosaics”, Junhwan Kim and Fabio Pellacini, 2002.
- “Simulated Decorative Mosaics”, Alejo Hausner, 2001.
- Photomosaics, Robert Silvers and Michael Hawley, 1997.
- "*Snake: Active contour model*", M. Kass, A. Witkin and D.Terzopoulos, Int. J. Computer Vision, 1987.
- Philip Lau and Katia Hristova, Student Project implementation of JIM http://www.ic.sunysb.edu/Stu/pwlau/