Network flows for image processing
Part I: Binary optimization and Graph-cut

Jérôme Darbon

CNRS / CMLA-ENS Cachan
Mathematics Department UCLA

Variational Methods and Compressive Sensing in Imaging

Fields Institute, Toronto, May 2012

Acknowledgements

- Joint work with
  - M. Sigelle (Telecom ParisTech/ENST)

Context and motivations

- Image Processing as optimization problems
  - restoration

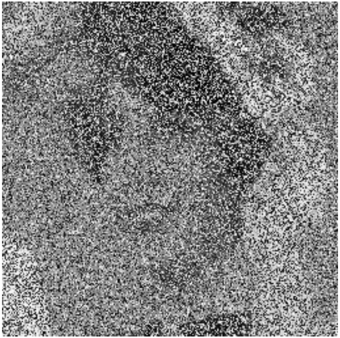
noisy image

restoration

image taken from [D. Sigelle 06]

Context and motivations

- Image Processing as optimization problems
  - restoration



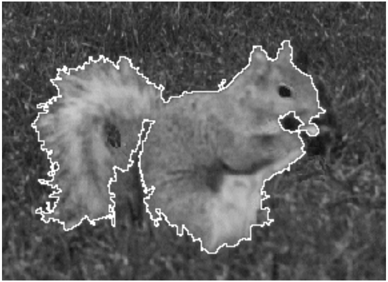
noisy image

restoration

image taken from [D. Sigelle 06]

- Image Processing as optimization problems
  - restoration, segmentation

original image



segmentation

image taken from [D. 05]

- Image Processing as optimization problems
  - restoration, segmentation

$$E(u|f,\lambda) = \underbrace{D(u,f)}_{\text{Data Fidelity}} + \lambda \underbrace{R(u)}_{\textit{a priori}/\text{Regularisation}}$$

- Several millions variables, can be non-convex
- Convex Continuous framework: Stopping criteria

$$E(u^{\epsilon}|v,\lambda) - E(u^{*},\lambda) \leq \epsilon \ .$$

$\rightarrow$ Optimal first-order approach [Nesterov 83,07], [Beck-et al 08],...
$\rightarrow$ Convergence in $O(\epsilon^{-1})$, $O(\epsilon^{-\frac{1}{2}})$
$\rightarrow$ non-polynomial ($\rightarrow \log \frac{1}{\epsilon}$)

- Refine the class of functionals
- Quid $\epsilon = 0$? (by definition: algorithm $\equiv$ finite number of iterations)

Context and motivations

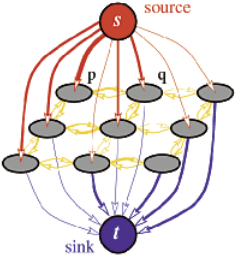
- Image Processing as optimization problems
  - restoration, segmentation

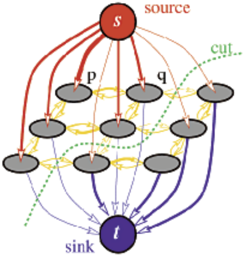
$$E(u|f,\lambda) = \underbrace{D(u,f)}_{\text{Data Fidelity}} + \lambda \underbrace{R(u)}_{\textit{a priori}/\text{Regularisation}}$$

- Several millions variables, generally non-convex
- Fast algorithm, and exact solutions for rigorous framework
  $\rightarrow$ [Winkler 03] Dissociation models/algorithms
- Discrete Framework $\rightarrow$ Markov Random Fields (MRFs)
  Optimization techniques : stochastic methods and combinatorics
  $\rightarrow$ energies formulated as a network flow

Context and motivations

- Combinatorics → exact optimization
  binary energies→ maximum flow/minimum-cut





- Fast algorithms for sparse graphs
- Seminal approach due to [Picard Ratliff Networks 75]
  - Focus on binary cases
  - → segmentation object/background with a perimeter prior
  - Used in Statistical physics in the 80's (Ferromagnetic Ising model)
  - Re-discovered by [Boykov *et al.* 01], . . . "Graph-cuts" and **extended**

→ Extension to non-binary cases

Outline of the Talks

1. Binary optimization and Graph-cut
2. Total Variation optimization and applications

Remarks:

- discrete world : finite number of labels
- finite dimension $\mathbb{R}^n$

1. Minimum-cuts in networks and interger programming
   - Definition: cuts, capacity, s-t minimum-cut, maximum-flow
   - maximum-flow / s,t mimum-cut duality
   - Ideas on algorithms for computing maximum-flows
   - Mapping binary optimizations to s-t minimum-cuts
   - Application to imaging : Ising Chan-Vese model

Maximum flow/Minimum cuts in networks: Definitions

- Consiger a graph (network) G = [V,A]
- $V = \{v_0, \dots, v_{n+1}\}$
- Directed arc from v_i to v_j with capacity c_{ij}
- Let v_0 and v_{n+1} represent the source and the sink, respectively

Definition

A cut separating v_0 and v_{n+1} is defined as a node partition $(S, \bar{S})$ where $v_0 \in S$, $v_{n+1} \in \bar{S}$, $S \cup \bar{S} = V$ and $S \cap \bar{S} = \emptyset$

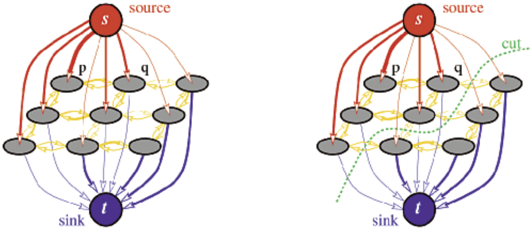
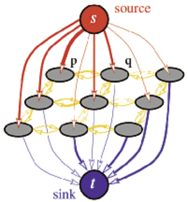
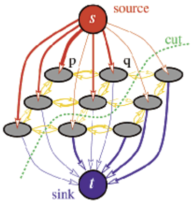


image taken from [Boykov et al PAMI 01]

Duality and maximum flows (1/3)





Definition

The capacity of a cut $C(S, \bar{S})$ can be defined as:

$$C(S, \bar{S}) = \sum_{i \in I} \sum_{j \in \bar{I}} c_{ij} \quad ,$$

where $I = \{i | v_i \in S\}$ *and* $\bar{I} = \{j | v_j \in \bar{S}\}$

- Goal: Minimize the capactity of the cut (s-t minimum-cut problem)

Duality and maximum flows (2/3)

- Goal: Minimize the capactity of the cut (s-t minimum-cut problem)
- Assumption: All the capacities are nonnegative
  $\Longrightarrow$ solved problem (polynomial time)

Max-flow/Min-cut Theorem (Duality)

The maximum value of the flow from a source node to a sink node in a capacitated network equals the minimum capacity among all s-t cuts

- Result independently discovered by
  - [Ford and Fulkerson 1956]
  - [Elias, Feinstein, Shannon 1956]

Duality and maximum flows (3/3)

- Computing maximum flows is a special linear program:

$$\begin{cases} \text{maximize } f \\ \text{s. t. } 0 \leq x_{ij} \leq c_{ij} \quad \leftarrow \text{feasibility of the flow} \\ \sum_{j:(i,j)\in A} x_{ij} + \sum_{j:(j,i)\in A} x_{ji} = \begin{cases} f & \text{for } i = v_0 \\ 0 & \text{for all } i \in V \setminus \{v_0, v_{n+1}\} \\ -f & \text{for all } i = v_{n+1} \end{cases} \end{cases}$$

the vector x is a flow and the value $f \in \mathbb{R}$ is the value of the flow.
Ideas for optimizing

- Maintain a feasible and "divergence free" flow
- Or maintain feasibility and allow to break "divergence free" constraint

Algorithms for computing maximum flows

- Assumption (recall): capacities are nonnegative
- Mainly two classes for computing maximum flows:
  - Augmenting Path class: augment flow along paths from source to sink while maintaining mass balance constraints.
  - Preflow-push class: flood the network so that some nodes have excesses. Send excess toward the sink or backward the source.
- Time complexity (n=# nodes, m=# arcs):
  - Labelling: $O(nmC)$
  - Sucessive shortest path: $O(n^2 m)$
  - FIFO preflow-push: $O(n^3)$
  - Highest preflow-push: $O(n^2\sqrt{m})$
  - Excess scaling: $O(nm + n^2 \log C)$

  where $C = \max_{ij} c_i j$

$\rightarrow$In practice time complexity is "quasi"-linear for "regular" graph using an augmenting-path based algorithm [Kolomogorov Boykov PAMI 03]

- Simple ideas on flows
  - Arc (i, j) has capacity c_{ij}
  - Suppose an arc carries x_{ij} units of flow
  - We can still send $c_{ij} - x_{ij}$ flow from i to j through (i, j)
  - We can send x_{ij} unit of flow from j to i
    i.e., we cancel the existing flow on the arc
- Residual graph
  - Given a flow x
  - the residual graph is defined as follows:
    - Replace each arc (i, j) in the original network by two arcs (i, j) and (j, i)
    - The arc (i,j) and residual capacity $r_{ij} = c_{ij} - x_{ij}$
    - The arc (j,i) and residual capacity $r_{ji} = x_{ij}$

Generic Augmenting Path algorithms

- Generic Algorithm
  - While there is a directed path from Source to Sink in residual graph
    - Identify an augmenting path P from Source to Sink
    - $\delta = \min\{r_{ij} : (i, j) \in P\}$
    - augment δ units of flow along P and compute residual graph
- Draw an example
- How to identify an augmenting path is important
  - for convergence toward the optimal
  - for time complexity
  - for image processing, use the [Kolomogorov-Boykov Pami 03] algorithm $\rightarrow$ quasi linear-time in practice

S-t minimum cuts and Binary Optimization (1/6)

- We follow the approach of [Picard and Ratliff 1975]
- As noted by [Hammer 1965], any cut separating v_0 and v_{n+1} can be represented by a vector

$$(1, x_1, x_2, \dots, x_n, 0)$$

where $x_j \in \{0, 1\}$ for $j = 1, 2, \dots, n$ and by defining $S = \{v_i | x_i = 1\}$ and $\bar{S} = \{v_i | x_i = 0\}$

- Every vector of the previous form represents some cut $(S, \bar{S})$.
- The capacity of a cut represented by $x_0 = 1$,$x_{n+1} = 0$ and $X = (x_1, \dots, x_n)$ can be represented as

$$C(X) = \sum_{i=0}^{n+1} \sum_{j=0}^{n+1} c_{ij} x_i (1 - x_j) \ ,$$

where $x_0 = 1$,$x_{n+1} = 0$.

S-t minimum cuts and Binary Optimization (2/6)

- Capacity of a cut (recall):

$$C(X) = \sum_{i=0}^{n+1} \sum_{j=0}^{n+1} c_{ij} x_i (1 - x_j) \ ,$$

- Now substitute $x_0 = 1$ and $x_{n+1} = 0$ and use that $x^2 = x$ for binary variables, we have:

$$\begin{aligned} C(X) \ &= \ \sum_{j=0}^{n+1} c_{0j} + \sum_{j=1}^{n+1} \left(c_{j,n+1} - c_{0j} + \sum_{i=1}^{n} c_{ji} \right) x_j \\ &\quad - \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} x_i x_j \end{aligned}$$

S-t minimum cuts and Binary Optimization (3/6)

- Now consider any boolean function of the form (recall)

$$F(X) = \sum_{j=1}^{n} p_j x_j - \sum_{i=1}^{n} \sum_{j=1}^{n} q_{ij} x_i x_j + K$$

Theorem [Picard and Ratliff 1975]

A network G with arc capacities c_{ij} satisfying

1. $c_{ij} + c_{ji} = q_{ij} + q_{ji}$ *for* $i, j = 1, \ldots, n$
2. $c_{j,n+1} - c_{0j} = p_j - \sum_{i=1}^{n} q_{ij}$ *for* $j = 1, \ldots, n$
3. $c_{0,n+1} = K - \sum_{j=1}^{n} c_{0j}$

has $C(X) = F(X)$ *for all X such that* $x_j \in \{0, 1\}$ *for j=1,...,n .*

- Minimizing F $\Leftrightarrow$ Finding a minimum s-t cut

- Recall that finding a minimum s-t cut is is
  - Polynomial when capacities are positive

Theorem [Picard and Ratliff 1975]

If in the binary energy F

$$F(X) = \sum_{j=1}^{n} p_j x_j - \sum_{i=1}^{n} \sum_{j=1}^{n} q_{ij} x_i x_j + K$$

we have $q_{ij} \geq 0$ for $i = 1, \ldots, n$ then one can build a network such that

- *the conditions of the previous th. are satisfied*
- *all capacities are nonnegative (polynomial time)*

- Statistical Phys.: MAP of Ferromagnetic Ising MRFs [Ogielsky 85]
- Binary Image restoration [Greig et al. 89]
- This is also called ”Graph-cut” [Boykov, Kolmogorov,... 01]

S-t minimum cuts and Binary Optimization (5/6)

- Thus we are able to solve exactly in polynomial time

$$\begin{cases} \text{minimize } \sum_{j=1}^{n} p_j x_j - \sum_{i=1}^{n} \sum_{j=1}^{n} q_{ij} x_i x_j \\ \text{s. t. } x_j \in \{0, 1\} \text{ for } j \in 1, \ldots, n \end{cases}$$

  where p_j and q_{ij} are some real valued constants and $q_{ij} \geq 0$.
- From a Bayesian point of view this a binary Markov Random Field (MRF) with pairwise interaction.
- In statistical physics this model is known as the ferromagnetic Ising model.

S-t minimum cuts and Binary Optimization (6/6)

- Application of the work of [Picard and Ratliff 1975]
  - [Barahona 1985] and [Ogielski 1986] studies the ground state of the Ising model from a stastical physics point of view.
  - [Greig *et al.* 1989] studies binary image restoration via the Ising model.
- This kind of approach has been revived by the re-introduction of combinatorial methods in image processing and computer vision.
- The graph-cut approach of Boykov-Veksler-Zabih goes far beyond since it copes with non-binary optimization.

How do we build the graph for

$$E(x) = \sum_{i=1}^{n} p_i x_i - \sum_{i=1}^{n} \sum_{j=1}^{n} q_{ij} x_i x_j$$

with $q_{ij} \geq 0$
We proceed term by term (Draw it)

- unary term: $p_i x_i$
  - case $p_i \geq 0$
  - case $p_i < 0$. $p_i x_i = -p_i(1 - x_i) + p_i$
  - capacity for (Source, i): $c_{0,i} = \max(0, c_i)$
  - capacity for (i, Sink): $c_{i,n+1} = \max(0, -c_i)$
- Pairwise term: $-q_{ij} x_i x_j$

How do we build the graph for

$$E(x) = \sum_{i=1}^{n} p_i x_i - \sum_{i=1}^{n} \sum_{j=1}^{n} q_{ij} x_i x_j$$

with $q_{ij} \geq 0$
We proceed term by term (Draw it)

- unary term: $p_i x_i$
  - case $p_i \geq 0$
  - case $p_i < 0$. $p_i x_i = -p_i(1 - x_i) + p_i$
  - capacity for (Source, i): $c_{0,i} = \max(0, c_i)$
  - capacity for (i, Sink): $c_{i,n+1} = \max(0, -c_i)$
- Pairwise term: $-q_{ij} x_i x_j$

How do we build the graph for

$$E(x) = \sum_{i=1}^{n} p_i x_i - \sum_{i=1}^{n} \sum_{j=1}^{n} q_{ij} x_i x_j$$

with $q_{ij} \geq 0$
We proceed term by term (Draw it)

- unary term: $p_i x_i$
  - case $p_i \geq 0$
  - case $p_i < 0$. $p_i x_i = -p_i(1 - x_i) + p_i$
  - capacity for (Source, i): $c_{0,i} = \max(0, c_i)$
  - capacity for (i, Sink): $c_{i,n+1} = \max(0, -c_i)$
- Pairwise term: $-q_{ij} x_i x_j$

Building the graph (2/2)

How do we build the graph for

$$E(x) = \sum_{i=1}^{n} p_i x_i - \sum_{i=1}^{n} \sum_{j=1}^{n} q_{ij} x_i x_j$$

with $q_{ij} \geq 0$
We proceed term by term (Draw it)

- Pairwise term: $-q_{ij} x_i x_j$
- Note that $\forall (x, y) \in \{0, 1\}^2 \ \ |x - y| = x + y - 2xy$

$\Rightarrow \ -q_{ij} x_i y_j = \frac{q_{ij}}{2} |x_i - x_j| - \frac{q_{ij}}{2} (x_i + x_j)$

Building the graph (2/2)

How do we build the graph for

$$E(x) = \sum_{i=1}^{n} p_i x_i - \sum_{i=1}^{n} \sum_{j=1}^{n} q_{ij} x_i x_j$$

with $q_{ij} \geq 0$

We proceed term by term (Draw it)

- Pairwise term: $-q_{ij} x_i x_j$
- Note that $\forall (x, y) \in \{0, 1\}^2 \ \ |x - y| = x + y - 2xy$

$\Rightarrow \ -q_{ij} x_i y_j = \frac{q_{ij}}{2} |x_i - x_j| - \frac{q_{ij}}{2} (x_i + x_j)$

Building the graph (2/2)

How do we build the graph for

$$E(x) = \sum_{i=1}^{n} p_i x_i - \sum_{i=1}^{n} \sum_{j=1}^{n} q_{ij} x_i x_j$$

with $q_{ij} \geq 0$

We proceed term by term (Draw it)

- Pairwise term: $-q_{ij} x_i x_j$
- Note that $\forall (x, y) \in \{0, 1\}^2 \ \ |x - y| = x + y - 2xy$

$\Rightarrow \ -q_{ij} x_i y_j = \frac{q_{ij}}{2} |x_i - x_j| - \frac{q_{ij}}{2} (x_i + x_j)$

Relation to "graph-cuts" and submodularity

Other combinatorial optimization schemes for binary energies

- "Graph-cuts" [Boykov, Veksler, Zabih PAMI 01]:
  Requires: $R(\alpha,\beta) \leq R(\alpha,\gamma) + R(\gamma,\beta)$ (triangle inequality)
- The latter is equivalent [Kolmogorov, Zabih PAMI 03] to:
  $R(0,0) + R(1,1) \leq R(0,1) + R(1,0)$ (*binary submodularity*)
  Necessary and sufficient condition

Proposition [D. DAM 09]

Assume E is a binary energy with pairwise interactions, i.e.,

$$E(x) = \sum_i f_i(x_i) + \sum_{(i,j)} g_{ij}(x_i, x_j) \ ,$$

with $\forall i\ x_i \in \{0,1\}$. Then E is exactly minimisable in polynomial time iff one of the following two equivalent assertions is satisfied:

- *[Picard et al.], all interactions write as $g_{ij}(x,y) = w_{ij}xy$ with $w_{st} \leq 0$,*
- *[Kolmogorov and Zabih], all pairwise interactions are submodular.*

Imaging Problems

- Using the reformulation, the regularization term takes the following form

$$\sum_i \sum_j w_{ij} |x_j - x_i|$$

- This can be seen as a discrete perimeter
- Draw picture

Imaging Problems: Markov Random Field (ISING)

- Binary segmentation (object/background)
- Solve for $u \in \{0,1\}^n$

$$
\begin{aligned}
E(u) &= \sum_j w_{ij}|u_j - u_i| \\
&\quad + \sum_i u_i D_i \;\leftarrow \text{Data term for assigning 1} \\
&\quad + \sum_i (1 - u_i) E_I \;\leftarrow \text{Data term for assigning 0}
\end{aligned} \tag{1}
$$

Chan-Vese Model/Active contours without edges

- The binary Mumford-Shah Model or Chan-Vese consists of approximating a signal with two constants with a prior on the boundary

$$
\begin{aligned}
E(\Omega_1, \mu_0, \mu_1 | v) \quad = \quad & \beta \, Per\,(\Omega_1) \\
& + \int_{\Omega \setminus \Omega_1} f(\mu_0, v(x)) dx \\
& + \int_{\Omega_1} f(\mu_1, v(x)) dx \quad ,
\end{aligned}
$$

- f is typically $\| \cdot \|_{L^p}^p$
- Issues:
  - Non-convex problem
  - Fast algorithm
  - Exact solution to:
    1. measure the quality of the model,
    2. measure the quality of an approximation algorithm

Chan-Vese Model: Naive Algorithm

- Discretization

$$
\begin{aligned}
E(u, \mu_0, \mu_1) &= \beta \sum_{(i,j)} w_{ij} |u_j - u_i| \\
&\quad + \sum_i (1 - u_i) \left\{ f(\mu_1, v_i) - f(\mu_0, v_i) \right\} \\
&\quad + \sum_i f(\mu_0, v_i) \ .
\end{aligned}
$$

- Set μ_0 and μ_1, $O(L^2)$ possible configurations
- Optimize for u via Graph-cut [Boykov et al. 01], [Picard Ratliff 75]

Chan-Vese Model: An inclusion Property

- Goal: reduce the number ($O(L^2)$) of maximum flows
- Idea: show an inclusion property of the solution
- Discretization (Recall)

$$
\begin{aligned}
E(u,\mu_0,\mu_1) &= \beta \sum_{(i,j)} w_{ij}|u_i - u_j| \\
&+ \sum_i (1-u_i)\left\{f(\mu_1,v_i) - f(\mu_0,v_i)\right\} \\
&+ \sum_i f(\mu_0,v_i) \ .
\end{aligned}
$$

- A variable which measures the difference between μ_0 and μ_1:

$$
\mu_1 = \mu_0 + K \ .
$$

- Thus we have

$$E(u, \mu_0, \mu_1) = \beta \sum_{(i,j)} w_{ij} |u_i - u_j| + \sum_i u_i \{ f(\mu_0, v_i) - f(\mu_0 + K, v_i) \} + Constant$$

- Assume K is fixed and define

$$E^k(u, \mu_0) = E(u, \mu_0, K)$$

Chan-Vese Model: An inclusion Property

- Assume K is fixed and define (Recall)

$$E^k(u,\mu_0) = E(u,\mu_0,K)$$

Theorem

Assume Data fidelity f is convex and assume $\widehat{\mu_0} \leq \widetilde{\mu_0}$. Let us defined the binary images $\widehat{u}$ and $\widetilde{u}$ as minimizers of $E^K(\cdot,\widehat{\mu_0})$ and $E^K(\cdot,\widetilde{\mu_0})$ respectively, i.e:

$$\widehat{u} \in \min\{u | E^K(u,\widehat{\mu_0})\} \ ,$$

$$\widetilde{u} \in \min\{u | E^K(u,\widetilde{\mu_0})\} \ .$$

Then we have the following inclusion:

$$\hat{u} \preceq \tilde{u} \ . \tag{2}$$

Chan-Vese Model: Algorithm

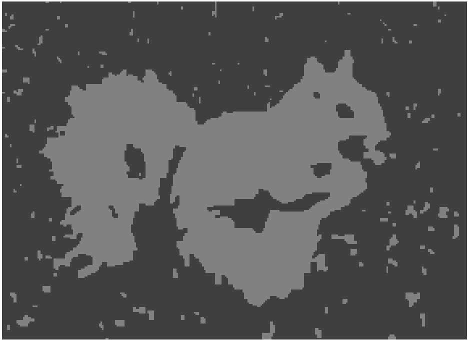
$\forall s \in S\ \hat{u}_s \leftarrow 0$
for $(K = 0; K < L; ++K)$
    Reset connected component map
    for $(\mu_0 = 0; (\mu_0 + K) < L; \mu_0 \leftarrow \mu_0 + 1)$
        $u' \leftarrow \underset{u}{\operatorname{argmin}}\ E^K(u, \mu_0)$
        if $(E^K(u', \mu_0) < E^K(\hat{u}, \mu_0))$
            $\hat{u} \leftarrow u'$
        update connected component map
return $\hat{u}$

Chan-Vese Model: Experiments

(Original)

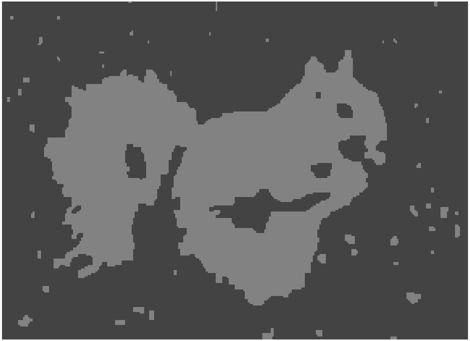
($\beta = 10$)

(a)

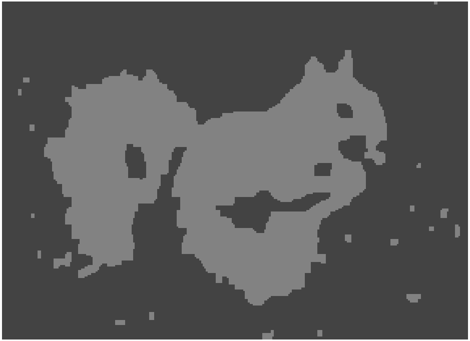


$\beta = 25$

Chan-Vese Model: Experiments



$\beta = 25$



$\beta = 30$

Chan-Vese Model: Time results

Time results in seconds (on a 3GHz Pentium IV) for *cameraman*
Direct approach in "$(\cdot)$"
And inclusion-based

Size	$\beta = 5$	$\beta = 10$	$\beta = 15$
32^2	4.16 (13.1)	4.4 (13.8)	4.8 (14.6)
64^2	17.1 (54.3)	17.8 (57.5)	18.5 (60.7)
128^2	72.57(243.3)	77.2 (254.6)	81.1 (268.4)
256^2	364.8 (1813.4)	382.2 (1851.7)	414.3 (2081.6)

Note: each binary binary takes about 0.02s for a 256^2 image

Conclusion

- Binary optimzation using maximum-flows
  - We can deal with problems of the form Perimeter + unary recall term
  - polynomial time and linear time in practice
- How can we extend to non-binary problems?