# Evaluating HYCOM Using the Surface Frontal Field

Peter Cornillon, Penelope Howe and David Ullman

Graduate School of Oceanography/University of Rhode Island

HYCOM Meeting
24-26 April 2007

# Overview

## Objective

To evaluate HYCOM as configured for the $\frac{1}{12}^{\circ}$ North Atlantic runs.

# Overview

## Objective

To evaluate HYCOM as configured for the $\frac{1}{12}^{\circ}$ North Atlantic runs.

## Approach

Compare the probability density of finding a surface SST front in 16x16 pixel regions in MODIS data with that of finding a surface front in HYCOM output.

# Overview

## Objective

To evaluate HYCOM as configured for the $\frac{1}{12}^{\circ}$ North Atlantic runs.

## Approach

Compare the probability density of finding a surface SST front in 16x16 pixel regions in MODIS data with that of finding a surface front in HYCOM output.

## Assumption

That the surface frontal field is a good proxy for upper ocean dynamics.

# Outline

# Outline

# The Algorithm



- We use an edge detection algorithm developed at URI by Jean-François Cayula
- The algorithm operates at 2 levels
  - The region level:
  - The local level:

# The Algorithm



- We use an edge detection algorithm developed at URI by Jean-François Cayula
- The algorithm operates at 2 levels
  - The region level:
    - Histograms are generated for overlapping $32 \times 32$ pixel windows.
    - The presence of a warm and a cool distribution suggests the existence of a front.
    - The cohesion of the distributions establishes the existence of a front.
    - Pixels at the warmest local minimum between the two distributions define frontal seed values.
  - The local level:
    - Beginning with the seed values a contour following algorithm finds frontal segments.
    - A contour ends when the contour changes direction by more than $90^{\circ}$ in 5 pixels.
    - A frontal segment must have 10 pixels.

# The Algorithm



- We use an edge detection algorithm developed at URI by Jean-François Cayula
- The algorithm operates at 2 levels
  - The region level:
    - Histograms are generated for overlapping $32 \times 32$ pixel windows.
    - The presence of a warm and a cool distribution suggests the existence of a front.
    - The cohesion of the distributions establishes the existence of a front.
    - Pixels at the warmest local minimum between the two distributions define frontal seed values.
  - The local level:
    - Beginning with the seed values a contour following algorithm finds frontal segments.
    - A contour ends when the contour changes direction by more than $90^{\circ}$ in 5 pixels.
    - A frontal segment must have 10 pixels.

# The Algorithm



- We use an edge detection algorithm developed at URI by Jean-François Cayula
- The algorithm operates at 2 levels
  - The region level:
    - Histograms are generated for overlapping 32 $\times$ 32 pixel windows.
    - The presence of a warm and a cool distribution suggests the existence of a front.
    - The cohesion of the distributions establishes the existence of a front.
    - Pixels at the warmest local minimum between the two distributions define frontal seed values.
  - The local level:
    - Beginning with the seed values a contour following algorithm finds frontal segments.
    - A contour ends when the contour changes direction by more than $90^{\circ}$ in 5 pixels.
    - A frontal segment must have 10 pixels.

# The Algorithm



- We use an edge detection algorithm developed at URI by Jean-François Cayula
- The algorithm operates at 2 levels
  - The region level:
    - Histograms are generated for overlapping 32 $\times$ 32 pixel windows.
    - The presence of a warm and a cool distribution suggests the existence of a front.
    - The cohesion of the distributions establishes the existence of a front.
    - Pixels at the warmest local minimum between the two distributions define frontal seed values.
  - The local level:
    - Beginning with the seed values a contour following algorithm finds frontal segments.
    - A contour ends when the contour changes direction by more than 90° in 5 pixels.
    - A frontal segment must have 10 pixels.

# The Algorithm



- We use an edge detection algorithm developed at URI by Jean-François Cayula
- The algorithm operates at 2 levels
  - The region level:
    - Histograms are generated for overlapping 32 $\times$ 32 pixel windows.
    - The presence of a warm and a cool distribution suggests the existence of a front.
    - The cohesion of the distributions establishes the existence of a front.
    - Pixels at the warmest local minimum between the two distributions define frontal seed values.
  - The local level:
    - Beginning with the seed values a contour following algorithm finds frontal segments.
    - A contour ends when the contour changes direction by more than 90° in 5 pixels.
    - A frontal segment must have 10 pixels.

# The Algorithm



- We use an edge detection algorithm developed at URI by Jean-François Cayula
- The algorithm operates at 2 levels
  - The region level:
    - Histograms are generated for overlapping 32 $\times$ 32 pixel windows.
    - The presence of a warm and a cool distribution suggests the existence of a front.
    - The cohesion of the distributions establishes the existence of a front.
    - Pixels at the warmest local minimum between the two distributions define frontal seed values.
  - The local level:
    - Beginning with the seed values a contour following algorithm finds frontal segments.
    - A contour ends when the contour changes direction by more than 90° in 5 pixels.
    - A frontal segment must have 10 pixels.

# The Algorithm



- We use an edge detection algorithm developed at URI by Jean-François Cayula
- The algorithm operates at 2 levels
  - The region level:
    - Histograms are generated for overlapping 32 $\times$ 32 pixel windows.
    - The presence of a warm and a cool distribution suggests the existence of a front.
    - The cohesion of the distributions establishes the existence of a front.
    - Pixels at the warmest local minimum between the two distributions define frontal seed values.
  - The local level:
    - Beginning with the seed values a contour following algorithm finds frontal segments.
    - A contour ends when the contour changes direction by more than 90$^\circ$ in 5 pixels.
    - A frontal segment must have 10 pixels.

# The Algorithm



- We use an edge detection algorithm developed at URI by Jean-François Cayula
- The algorithm operates at 2 levels
  - The region level:
    - Histograms are generated for overlapping 32 $\times$ 32 pixel windows.
    - The presence of a warm and a cool distribution suggests the existence of a front.
    - The cohesion of the distributions establishes the existence of a front.
    - Pixels at the warmest local minimum between the two distributions define frontal seed values.
  - The local level:
    - Beginning with the seed values a contour following algorithm finds frontal segments.
    - A contour ends when the contour changes direction by more than 90$^\circ$ in 5 pixels.
    - A frontal segment must have 10 pixels.

# The Algorithm



- We use an edge detection algorithm developed at URI by Jean-François Cayula
- The algorithm operates at 2 levels
  - The region level:
    - Histograms are generated for overlapping 32 $\times$ 32 pixel windows.
    - The presence of a warm and a cool distribution suggests the existence of a front.
    - The cohesion of the distributions establishes the existence of a front.
    - Pixels at the warmest local minimum between the two distributions define frontal seed values.
  - The local level:
    - Beginning with the seed values a contour following algorithm finds frontal segments.
    - A contour ends when the contour changes direction by more than 90$^\circ$ in 5 pixels.
    - A frontal segment must have 10 pixels.

# The Algorithm



- We use an edge detection algorithm developed at URI by Jean-François Cayula
- The algorithm operates at 2 levels
  - The region level:
    - Histograms are generated for overlapping 32 $\times$ 32 pixel windows.
    - The presence of a warm and a cool distribution suggests the existence of a front.
    - The cohesion of the distributions establishes the existence of a front.
    - Pixels at the warmest local minimum between the two distributions define frontal seed values.
  - The local level:
    - Beginning with the seed values a contour following algorithm finds frontal segments.
    - A contour ends when the contour changes direction by more than 90$^\circ$ in 5 pixels.
    - A frontal segment must have 10 pixels.

# The Probability Fields



- The statistic that we use to evaluate HYCOM is the probability of finding a front in a given region
- The frontal probability is defined by:

$$\frac{\sum_{i=1}^{N} \text{Front}_i}{\sum_{i=1}^{N} \text{Clear}_i}$$

  - $i$ is over all pixels in a spatial and temporal region,
  - $\text{Front}_i$ is 1 if the pixel is a front pixel, 0 otherwise and
  - $\text{Clear}_i$ is 1 if the pixel is clear, 0 otherwise.
- To obtain good statistics, probabilities are calculated for:
  - $16 \times 16$ pixel regions
  - For all fields in 2004; one per day.
  - I.e., $N = 16 \times 16 \times 366 = 93,696$.

# The Probability Fields



- The statistic that we use to evaluate HYCOM is the probability of finding a front in a given region
- The frontal probability is defined by:

$$\frac{\sum_{i=1}^{N} \text{Front}_i}{\sum_{i=1}^{N} \text{Clear}_i}$$

  - $i$ is over all pixels in a spatial and temporal region,
  - $\text{Front}_i$ is 1 if the pixel is a front pixel, 0 otherwise and
  - $\text{Clear}_i$ is 1 if the pixel is clear, 0 otherwise.
- To obtain good statistics, probabilities are calculated for:
  - $16 \times 16$ pixel regions
  - For all fields in 2004; one per day.
  - I.e., $N = 16 \times 16 \times 366 = 93,696$.

# The Probability Fields



- The statistic that we use to evaluate HYCOM is the probability of finding a front in a given region
- The frontal probability is defined by:

$$\frac{\sum_{i=1}^{N} \text{Front}_i}{\sum_{i=1}^{N} \text{Clear}_i}$$

  - $i$ is over all pixels in a spatial and temporal region,
  - $\text{Front}_i$ is 1 if the pixel is a front pixel, 0 otherwise and
  - $\text{Clear}_i$ is 1 if the pixel is clear, 0 otherwise.
- To obtain good statistics, probabilities are calculated for:
  - $16 \times 16$ pixel regions
  - For all fields in 2004; one per day.
  - I.e., $N = 16 \times 16 \times 366 = 93,696$.

# Outline

# The First Pass

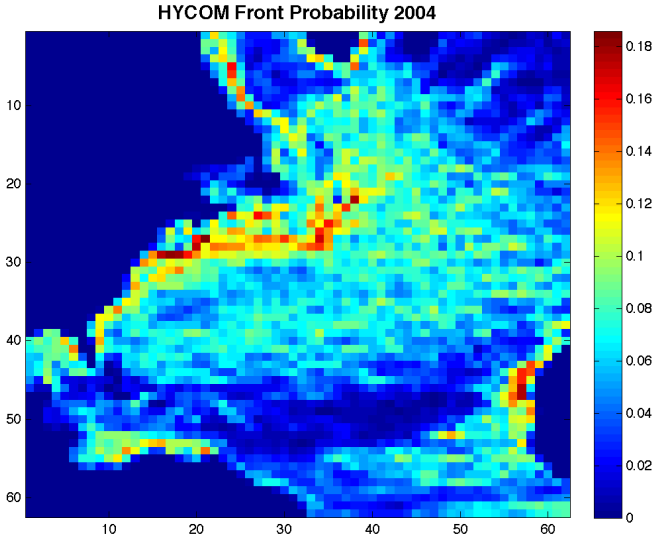


We began by calculating frontal probability for HYCOM

# The First Pass



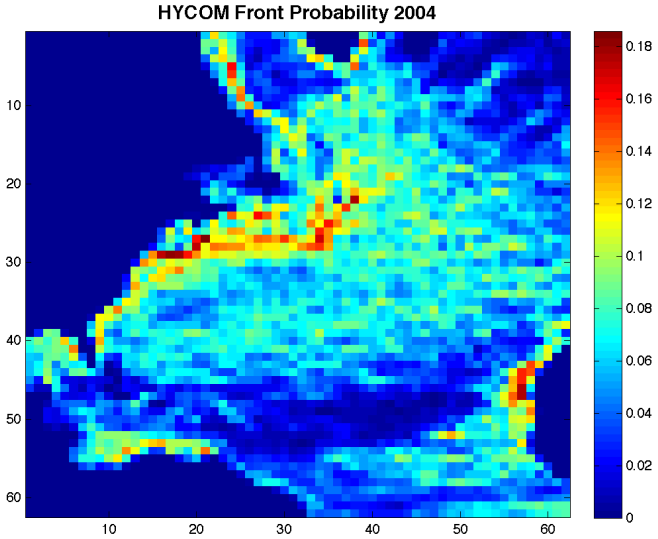
We began by calculating frontal probability for HYCOM

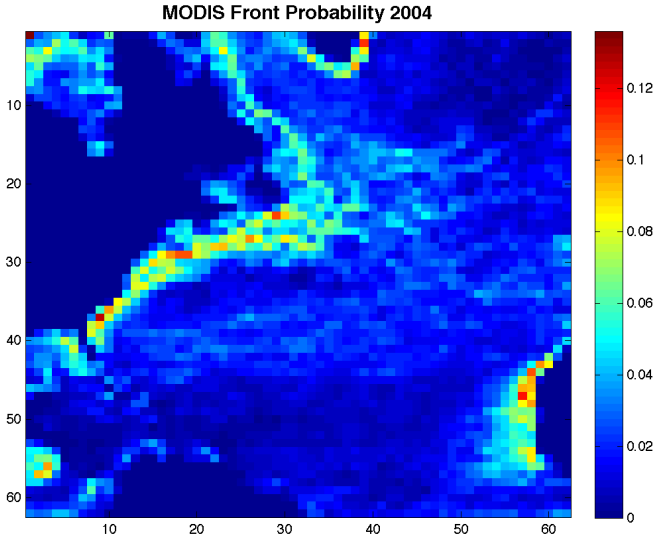


# The First Pass



We began by calculating frontal probability for HYCOM and MODIS 2004

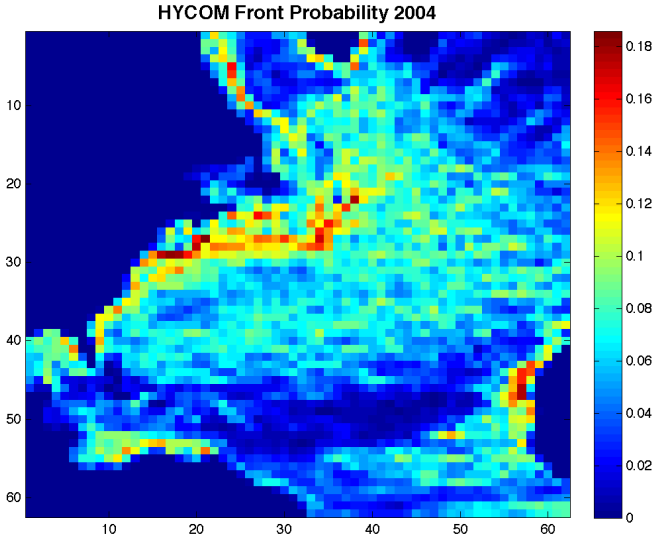


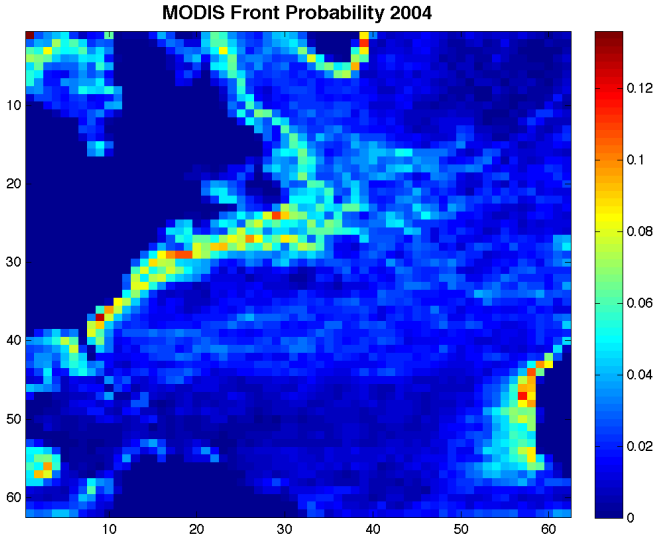



# The First Pass



We began by calculating frontal probability for HYCOM and MODIS 2004




Some similarities, but big differences, especially in the center of the gyre.

# Clouds Effect on Edge Detection

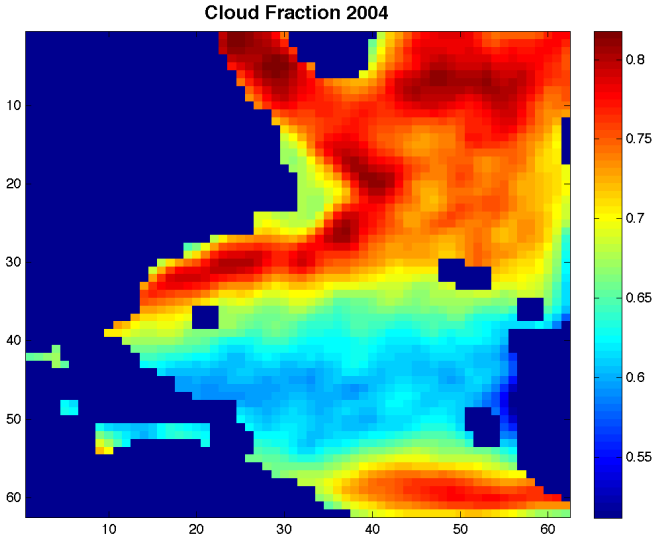


- The cloud field varies significantly over this region.
- We correct for cloudiness by dividing by the number of clear pixles,
- But the edge detection algorithm relies on histograms
- To calculate a histogram
  - More than 10% of the pixels must be clear, and
  - At least 25% of the clear pixels must be in smallest population.
  - Many 16$x$16 pixel regions with some clear pixels will fail these tests.
- Clouds will result in fronts not being detected at clear pixel locations.
- The fronts found are still good, but the algorithm underestimates probabilities.

# Clouds Effect on Edge Detection



- The cloud field varies significantly over this region.

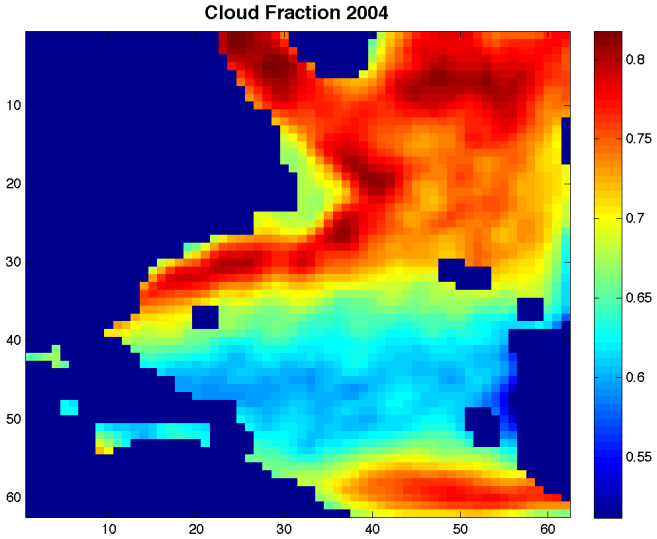


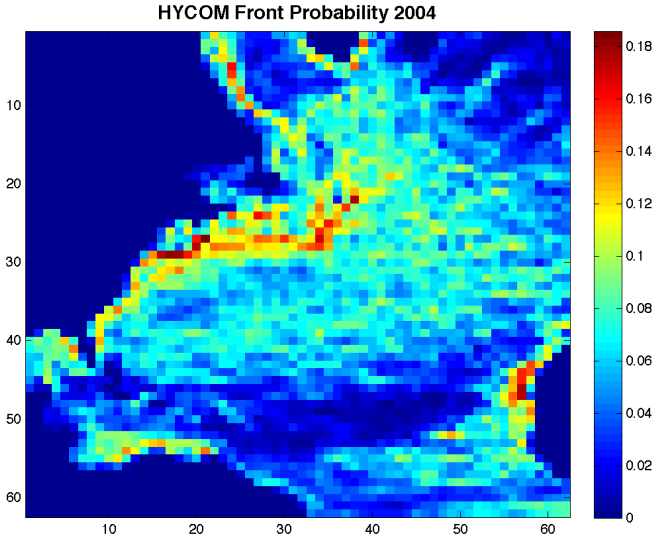

- We correct for cloudiness by dividing by the number of clear pixles,
- But the edge detection algorithm relies on histograms
- To calculate a histogram
  - More than 10% of the pixels must be clear, and
  - At least 25% of the clear pixels must be in smallest population.
  - Many 16x16 pixel regions with some clear pixels will fail these tests.
- Clouds will result in fronts not being detected at clear pixel locations.
- The fronts found are still good, but the algorithm underestimates probabilities.

# Clouds Effect on Edge Detection



- The cloud field varies significantly over this region.



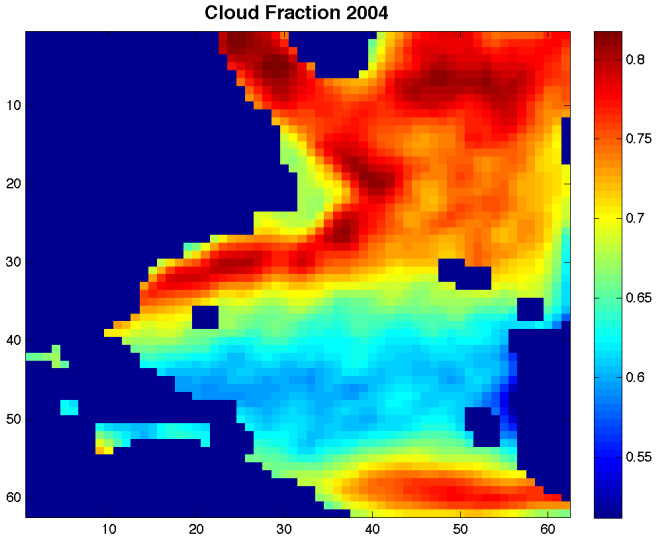



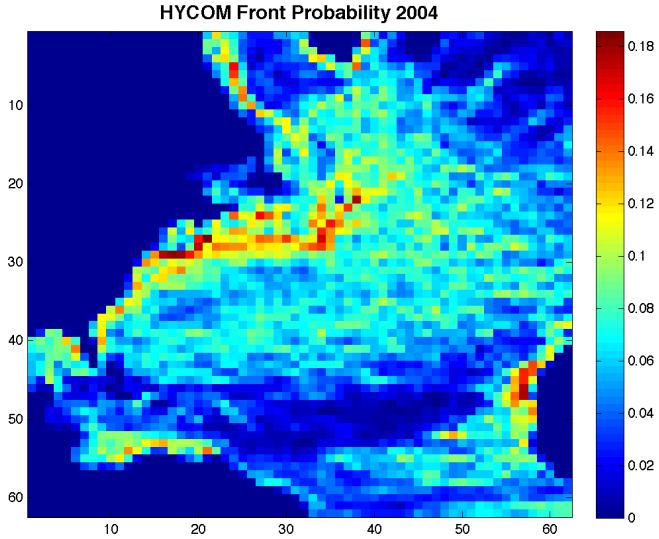

- We correct for cloudiness by dividing by the number of clear pixles,
- But the edge detection algorithm relies on histograms
- To calculate a histogram
  - More than 10% of the pixels must be clear, and
  - At least 25% of the clear pixels must be in smallest population.
  - Many 16x16 pixel regions with some clear pixels will fail these tests.
- Clouds will result in fronts not being detected at clear pixel locations.
- The fronts found are still good, but the algorithm underestimates probabilities.

# Clouds Effect on Edge Detection



- The cloud field varies significantly over this region.



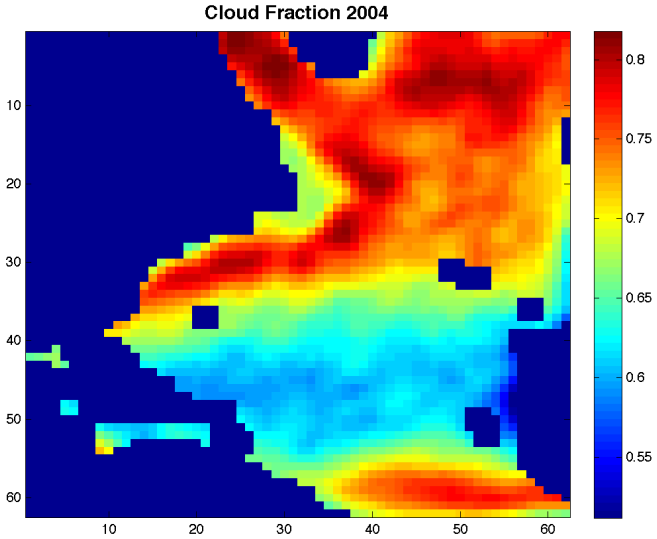



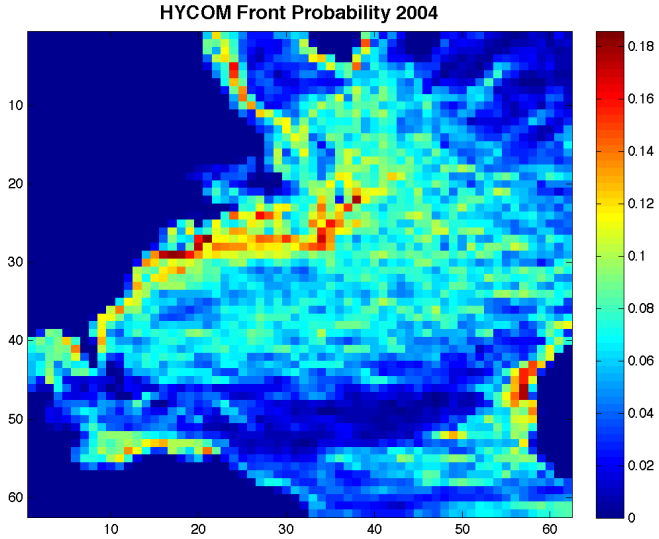

- We correct for cloudiness by dividing by the number of clear pixles,
- But the edge detection algorithm relies on histograms
- To calculate a histogram
  - More than 10% of the pixels must be clear, and
  - At least 25% of the clear pixels must be in smallest population.
  - Many 16x16 pixel regions with some clear pixels will fail these tests.
- Clouds will result in fronts not being detected at clear pixel locations.
- The fronts found are still good, but the algorithm underestimates probabilities.

# Clouds Effect on Edge Detection



- The cloud field varies significantly over this region.



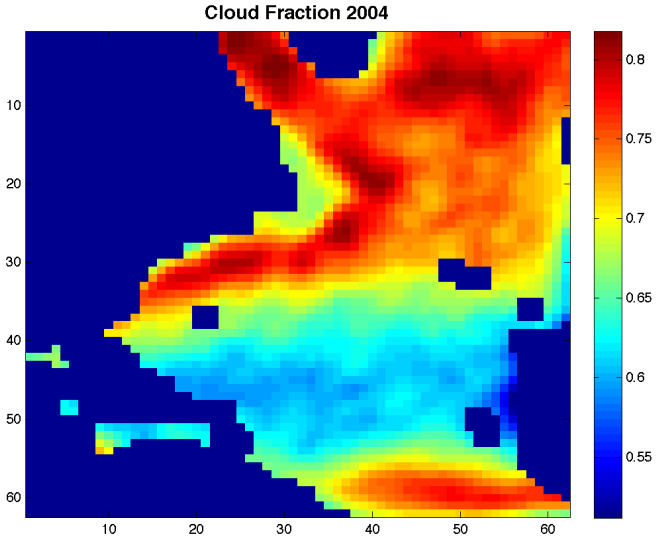



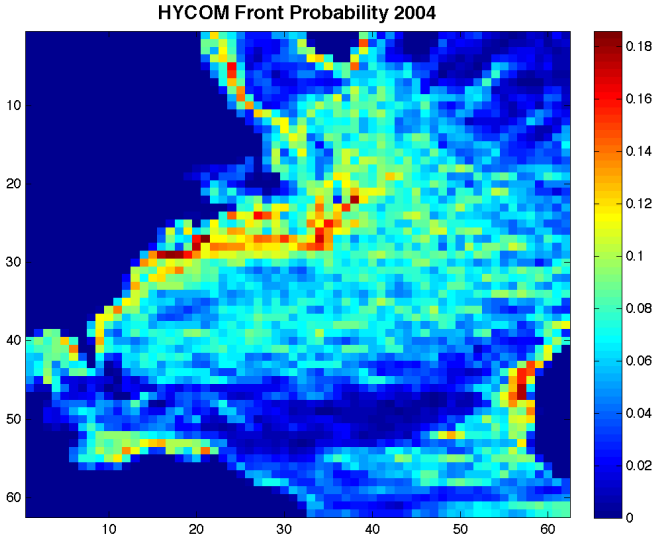

- We correct for cloudiness by dividing by the number of clear pixles,
- But the edge detection algorithm relies on histograms
- To calculate a histogram
  - More than 10% of the pixels must be clear, and
  - At least 25% of the clear pixels must be in smallest population.
  - Many 16x16 pixel regions with some clear pixels will fail these tests.
- Clouds will result in fronts not being detected at clear pixel locations.
- The fronts found are still good, but the algorithm underestimates probabilities.

# Clouds Effect on Edge Detection



- The cloud field varies significantly over this region.



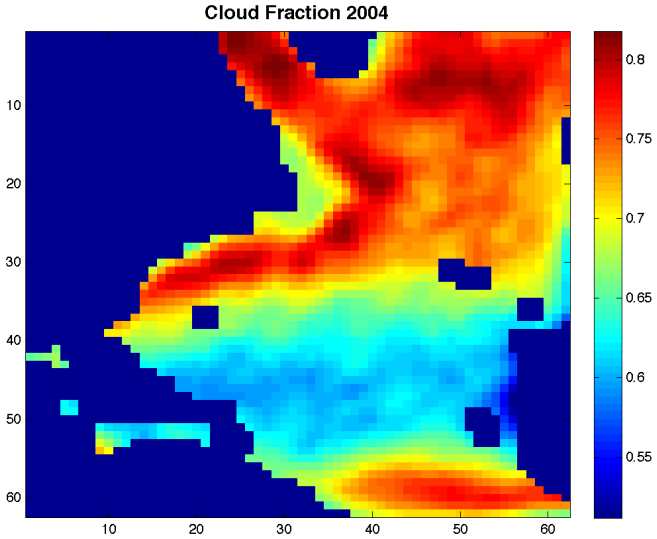



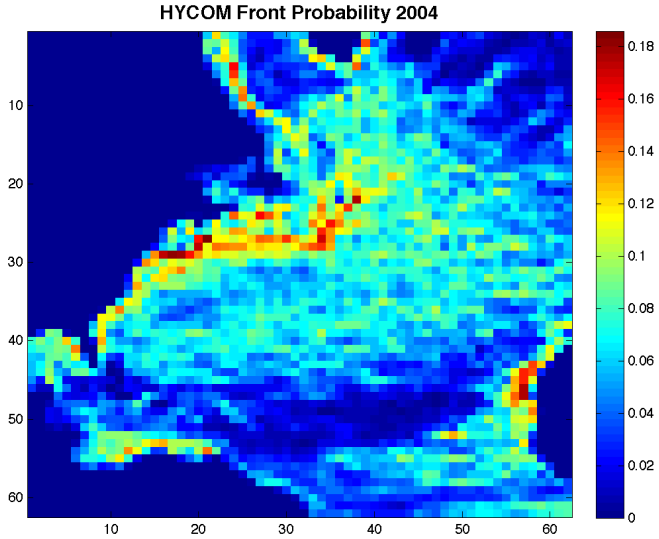

- We correct for cloudiness by dividing by the number of clear pixles,
- But the edge detection algorithm relies on histograms
- To calculate a histogram
  - More than 10% of the pixels must be clear, and
  - At least 25% of the clear pixels must be in smallest population.
  - Many 16$x$16 pixel regions with some clear pixels will fail these tests.
- Clouds will result in fronts not being detected at clear pixel locations.
- The fronts found are still good, but the algorithm underestimates probabilities.

# Clouds Effect on Edge Detection



- The cloud field varies significantly over this region.



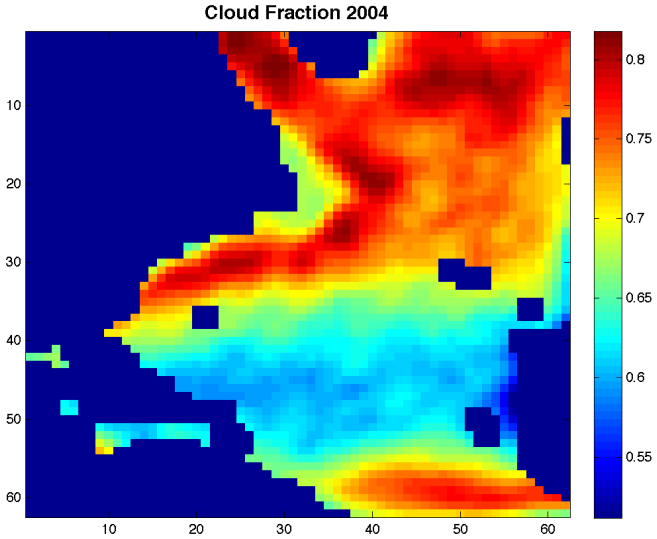



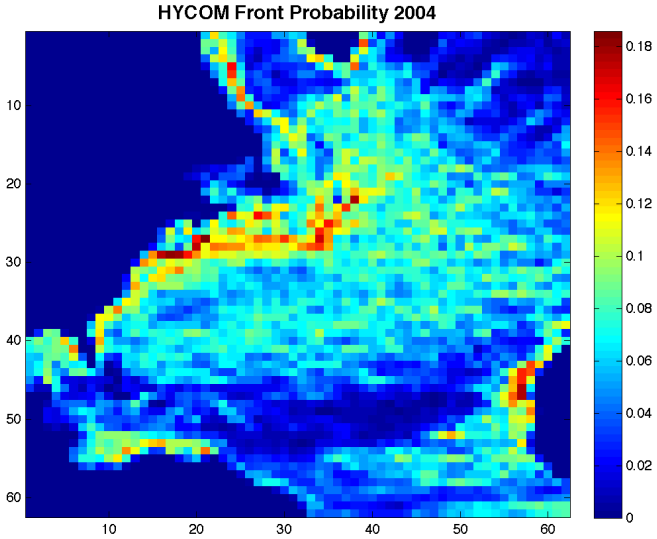

- We correct for cloudiness by dividing by the number of clear pixles,
- But the edge detection algorithm relies on histograms
- To calculate a histogram
  - More than 10% of the pixels must be clear, and
  - At least 25% of the clear pixels must be in smallest population.
  - Many 16$x$16 pixel regions with some clear pixels will fail these tests.
- Clouds will result in fronts not being detected at clear pixel locations.
- The fronts found are still good, but the algorithm underestimates probabilities.

# Clouds Effect on Edge Detection



- The cloud field varies significantly over this region.



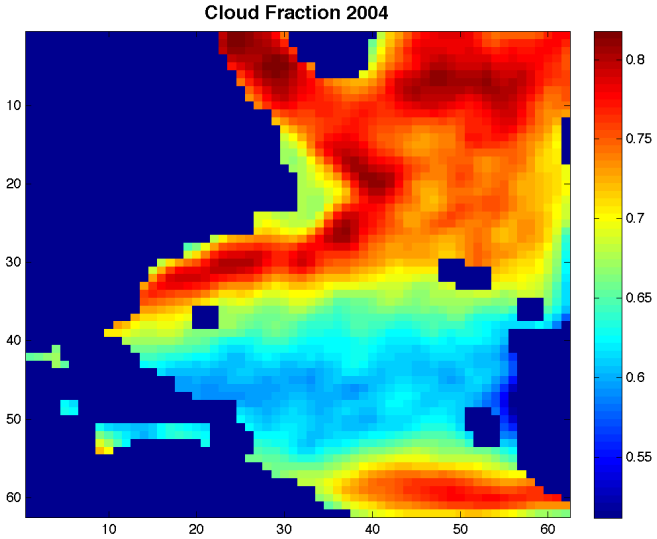



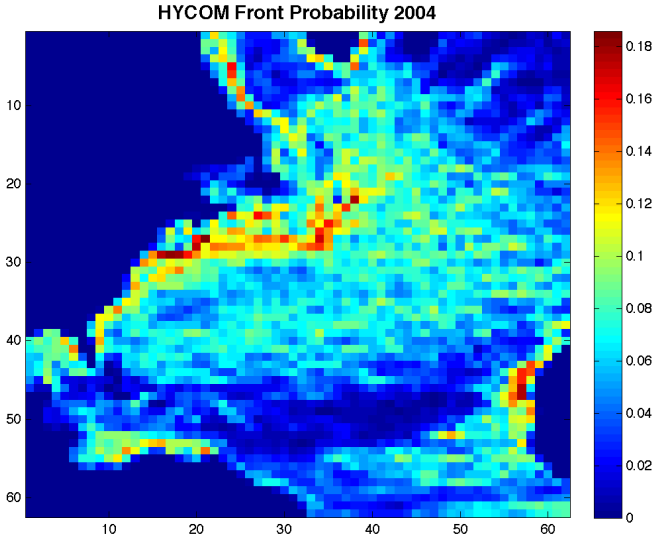

- We correct for cloudiness by dividing by the number of clear pixles,
- But the edge detection algorithm relies on histograms
- To calculate a histogram
  - More than 10% of the pixels must be clear, and
  - At least 25% of the clear pixels must be in smallest population.
  - Many 16$x$16 pixel regions with some clear pixels will fail these tests.
- Clouds will result in fronts not being detected at clear pixel locations.
- The fronts found are still good, but the algorithm underestimates probabilities.

# Clouds Effect on Edge Detection



- The cloud field varies significantly over this region.



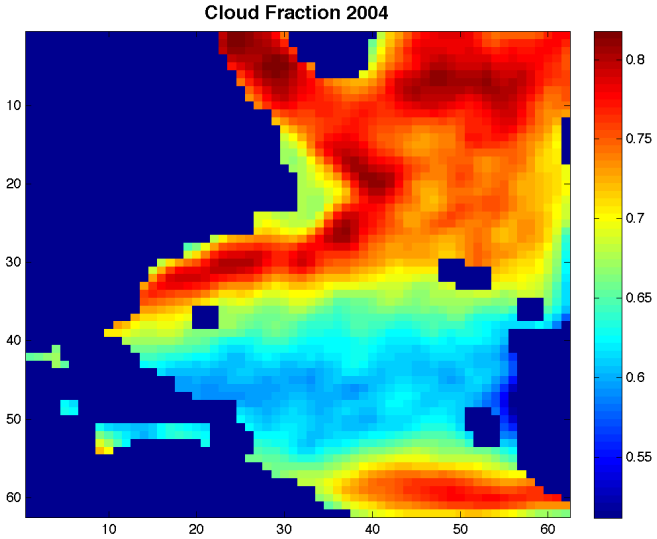



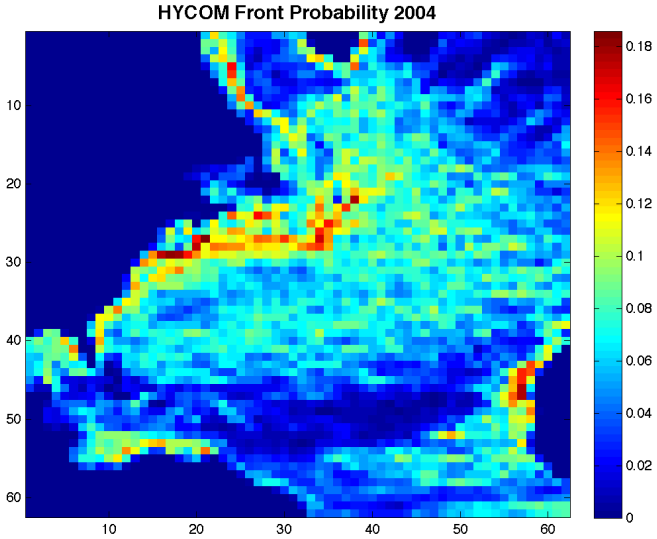

- We correct for cloudiness by dividing by the number of clear pixles,
- But the edge detection algorithm relies on histograms
- To calculate a histogram
  - More than 10% of the pixels must be clear, and
  - At least 25% of the clear pixels must be in smallest population.
  - Many 16$x$16 pixel regions with some clear pixels will fail these tests.
- Clouds will result in fronts not being detected at clear pixel locations.
- The fronts found are still good, but the algorithm underestimates probabilities.

# Clouds Effect on Edge Detection



- The cloud field varies significantly over this region.



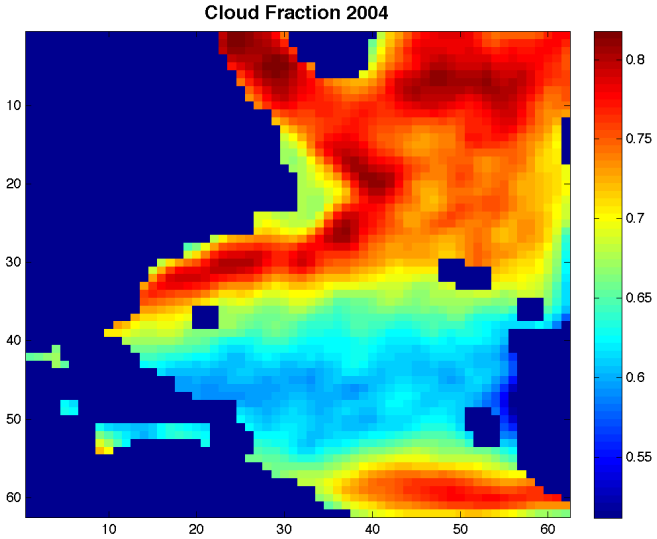



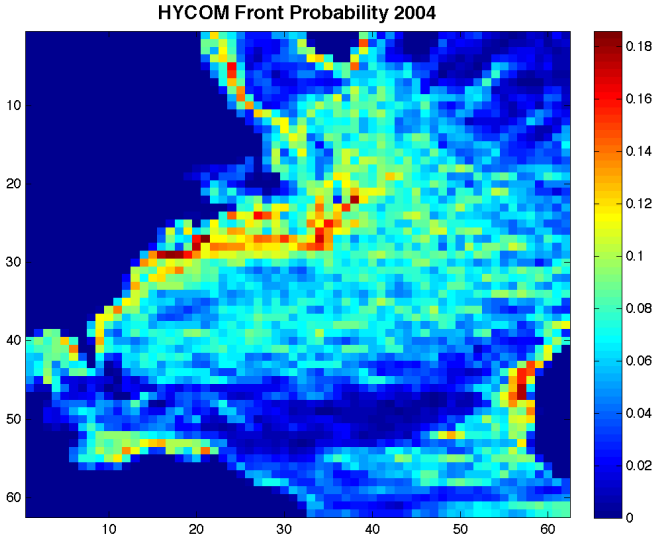

- We correct for cloudiness by dividing by the number of clear pixles,
- But the edge detection algorithm relies on histograms
- To calculate a histogram
  - More than 10% of the pixels must be clear, and
  - At least 25% of the clear pixels must be in smallest population.
  - Many 16$x$16 pixel regions with some clear pixels will fail these tests.
- Clouds will result in fronts not being detected at clear pixel locations.
- The fronts found are still good, but the algorithm underestimates probabilities.

# Clouds Effect on Edge Detection



- The cloud field varies significantly over this region.




- We correct for cloudiness by dividing by the number of clear pixles,
- But the edge detection algorithm relies on histograms
- To calculate a histogram
  - More than 10% of the pixels must be clear, and
  - At least 25% of the clear pixels must be in smallest population.
  - Many 16$x$16 pixel regions with some clear pixels will fail these tests.
- Clouds will result in fronts not being detected at clear pixel locations.
- The fronts found are still good, but the algorithm underestimates probabilities.

# Outline

# Calculate Probabilities Imposing Clouds on HYCOM



- Simply calculate fronts obtained from HYCOM with the MODIS clouds imposed on SST field, $F_{Cloudy}$.
- Map MODIS clouds onto HYCOM SST images.
- Run the edge detector on the 'cloudy' HYCOM fields.
- Compute the front probability field from the results, $F_{Cloudy}$.
- Compare $F_{Cloudy}$

# Calculate Probabilities Imposing Clouds on HYCOM



- Simply calculate fronts obtained from HYCOM with the MODIS clouds imposed on SST field, $F_{Cloudy}$.
- Map MODIS clouds onto HYCOM SST images.
- Run the edge detector on the 'cloudy' HYCOM fields.
- Compute the front probability field from the results, $F_{Cloudy}$.
- Compare $F_{Cloudy}$

# Calculate Probabilities Imposing Clouds on HYCOM



- Simply calculate fronts obtained from HYCOM with the MODIS clouds imposed on SST field, $F_{Cloudy}$.
- Map MODIS clouds onto HYCOM SST images.
- Run the edge detector on the 'cloudy' HYCOM fields.
- Compute the front probability field from the results, $F_{Cloudy}$.
- Compare $F_{Cloudy}$

# Calculate Probabilities Imposing Clouds on HYCOM

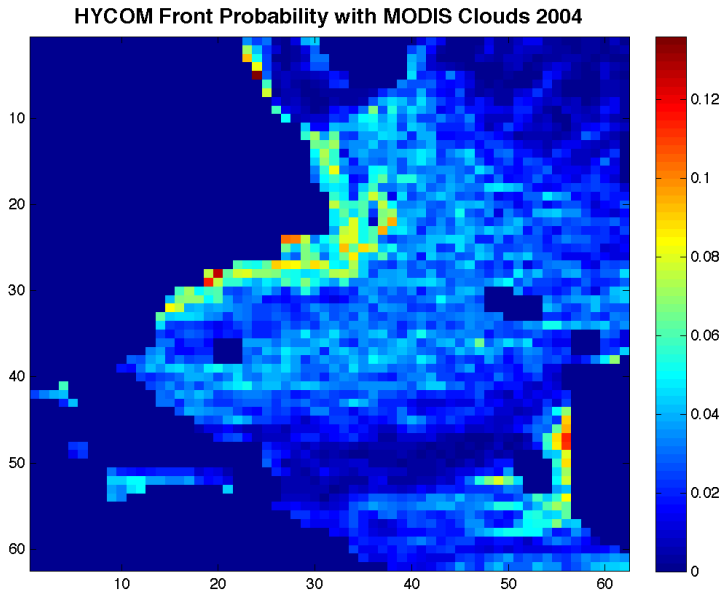


- Simply calculate fronts obtained from HYCOM with the MODIS clouds imposed on SST field, $F_{Cloudy}$.
- Map MODIS clouds onto HYCOM SST images.
- Run the edge detector on the 'cloudy' HYCOM fields.
- Compute the front probability field from the results, $F_{Cloudy}$.
- Compare $F_{Cloudy}$

# Calculate Probabilities Imposing Clouds on HYCOM



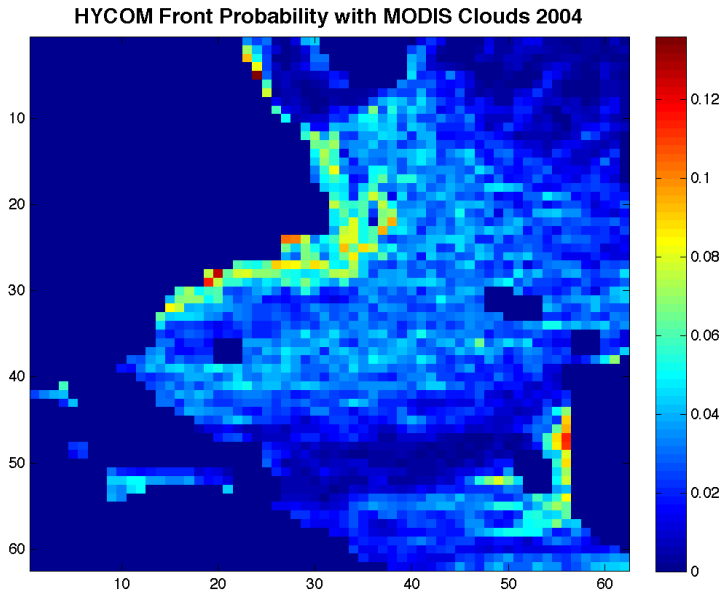
- Simply calculate fronts obtained from HYCOM with the MODIS clouds imposed on SST field, $F_{Cloudy}$.
- Map MODIS clouds onto HYCOM SST images.
- Run the edge detector on the 'cloudy' HYCOM fields.
- Compute the front probability field from the results, $F_{Cloudy}$.
- Compare $F_{Cloudy}$

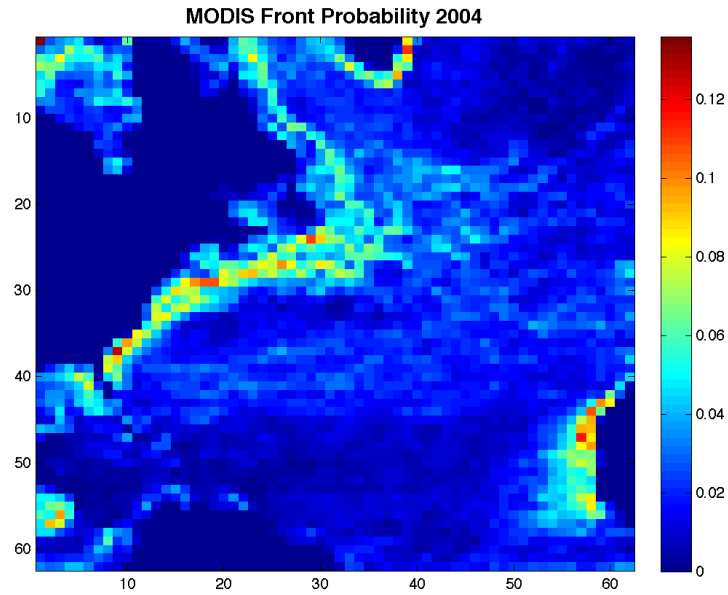


# Calculate Probabilities Imposing Clouds on HYCOM



- Simply calculate fronts obtained from HYCOM with the MODIS clouds imposed on SST field, $F_{Cloudy}$.
- Map MODIS clouds onto HYCOM SST images.
- Run the edge detector on the 'cloudy' HYCOM fields.
- Compute the front probability field from the results, $F_{Cloudy}$.
- Compare $F_{Cloudy}$ with $F_{MODIS}$

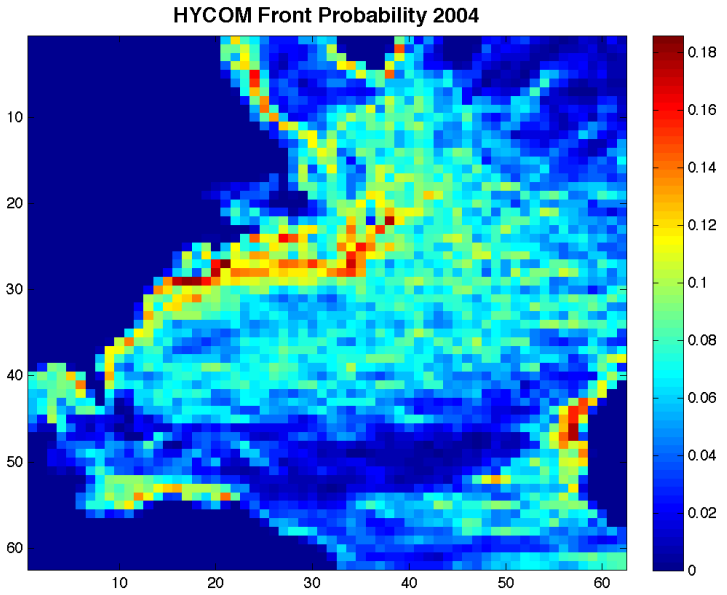


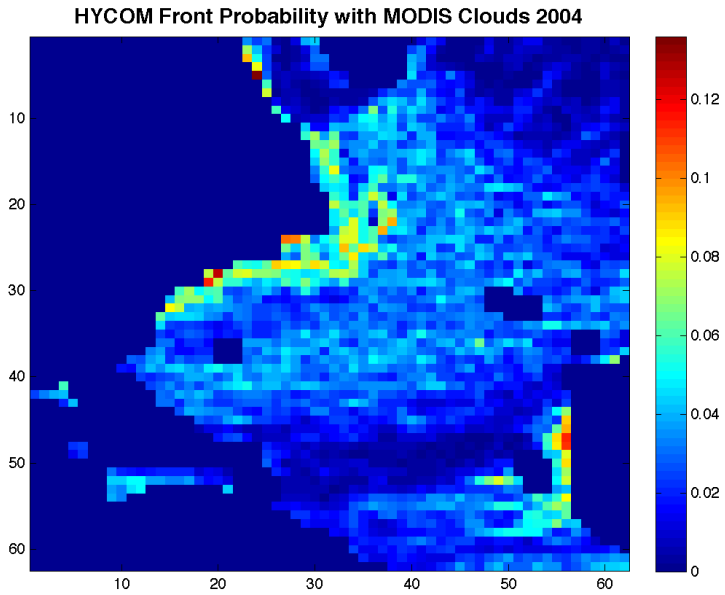



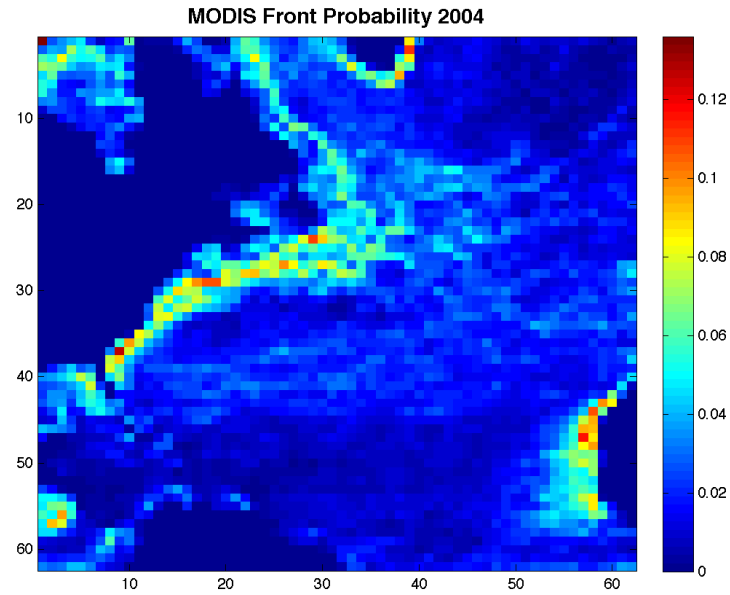
# Calculate Probabilities Imposing Clouds on HYCOM



- Simply calculate fronts obtained from HYCOM with the MODIS clouds imposed on SST field, $F_{Cloudy}$.
- Map MODIS clouds onto HYCOM SST images.
- Run the edge detector on the 'cloudy' HYCOM fields.
- Compute the front probability field from the results, $F_{Cloudy}$.
- Compare $F_{Cloudy}$ with $F_{MODIS}$



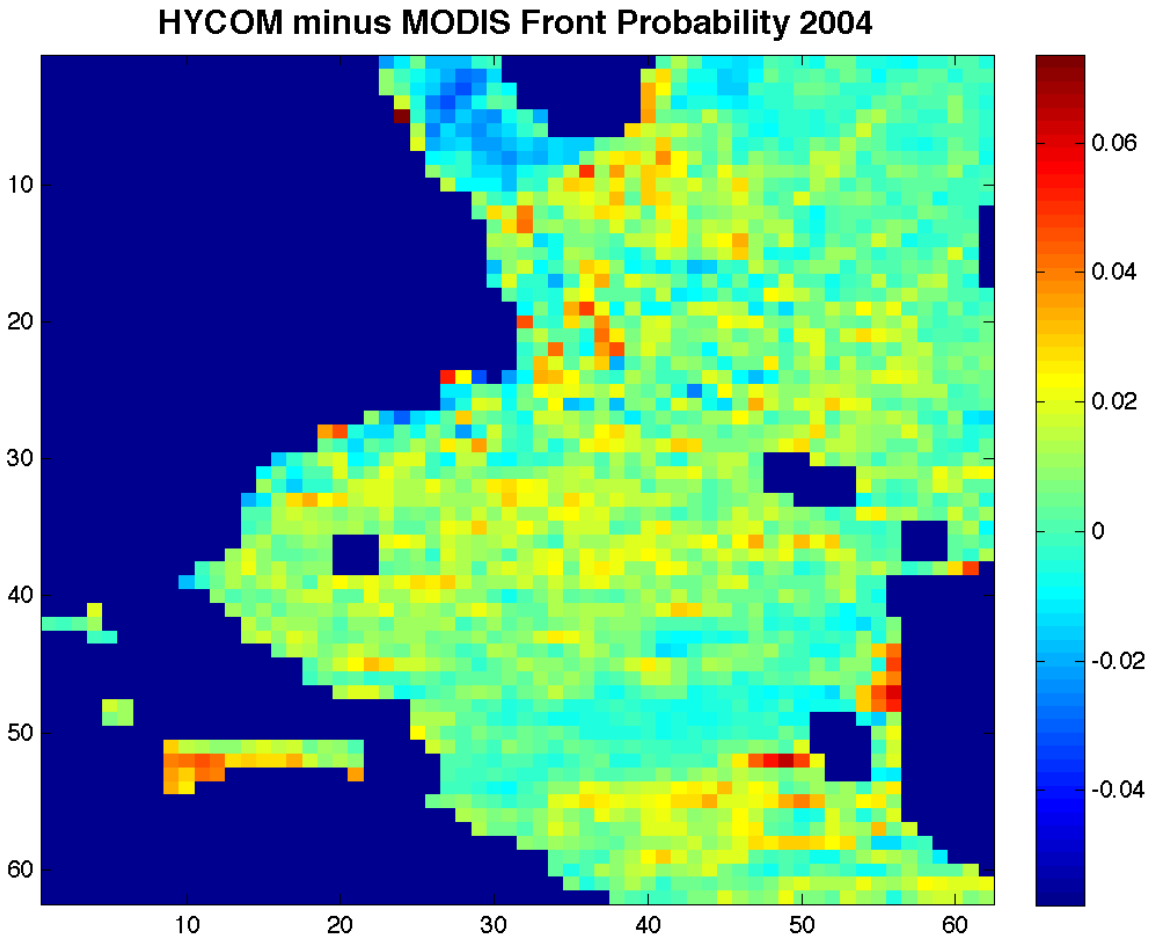



# Frontal Probability Differences



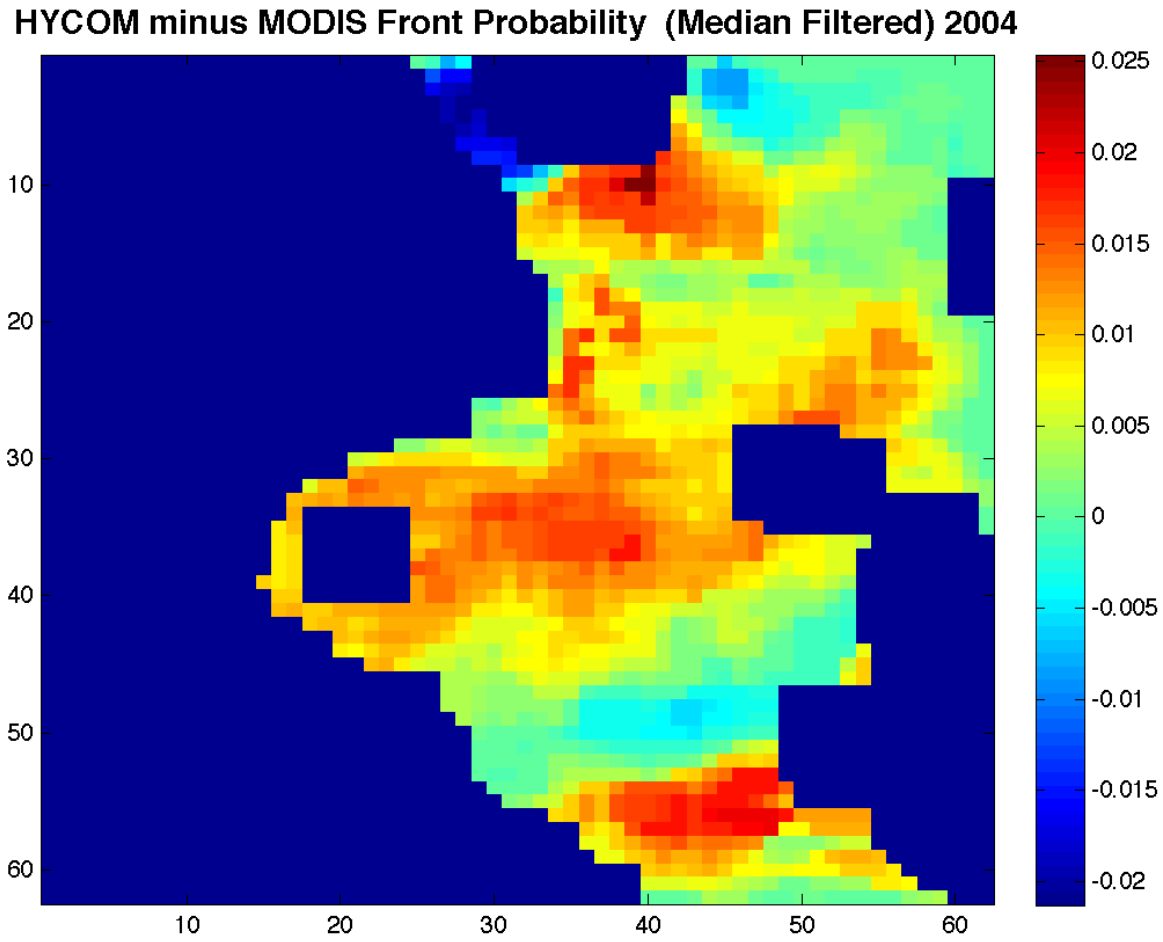
- Of more interest is the difference between the HYCOM frontal structure and that of MODiS.



# Frontal Probability Differences



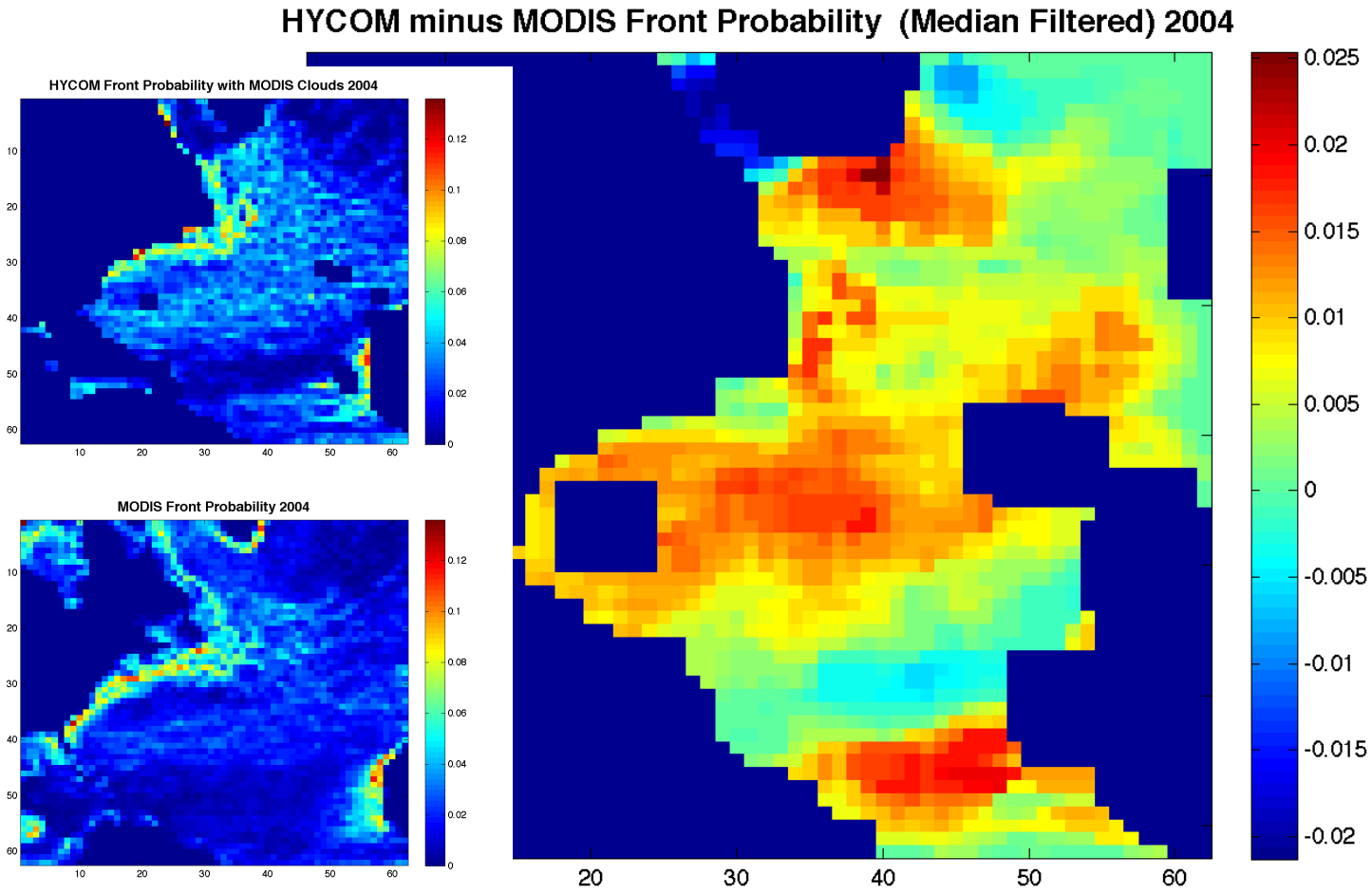
- Of more interest is the difference between the HYCOM frontal structure and that of MODiS. Median filtered.



# Frontal Probability Differences



- Of more interest is the difference between the HYCOM frontal structure and that of MODiS. Median filtered.

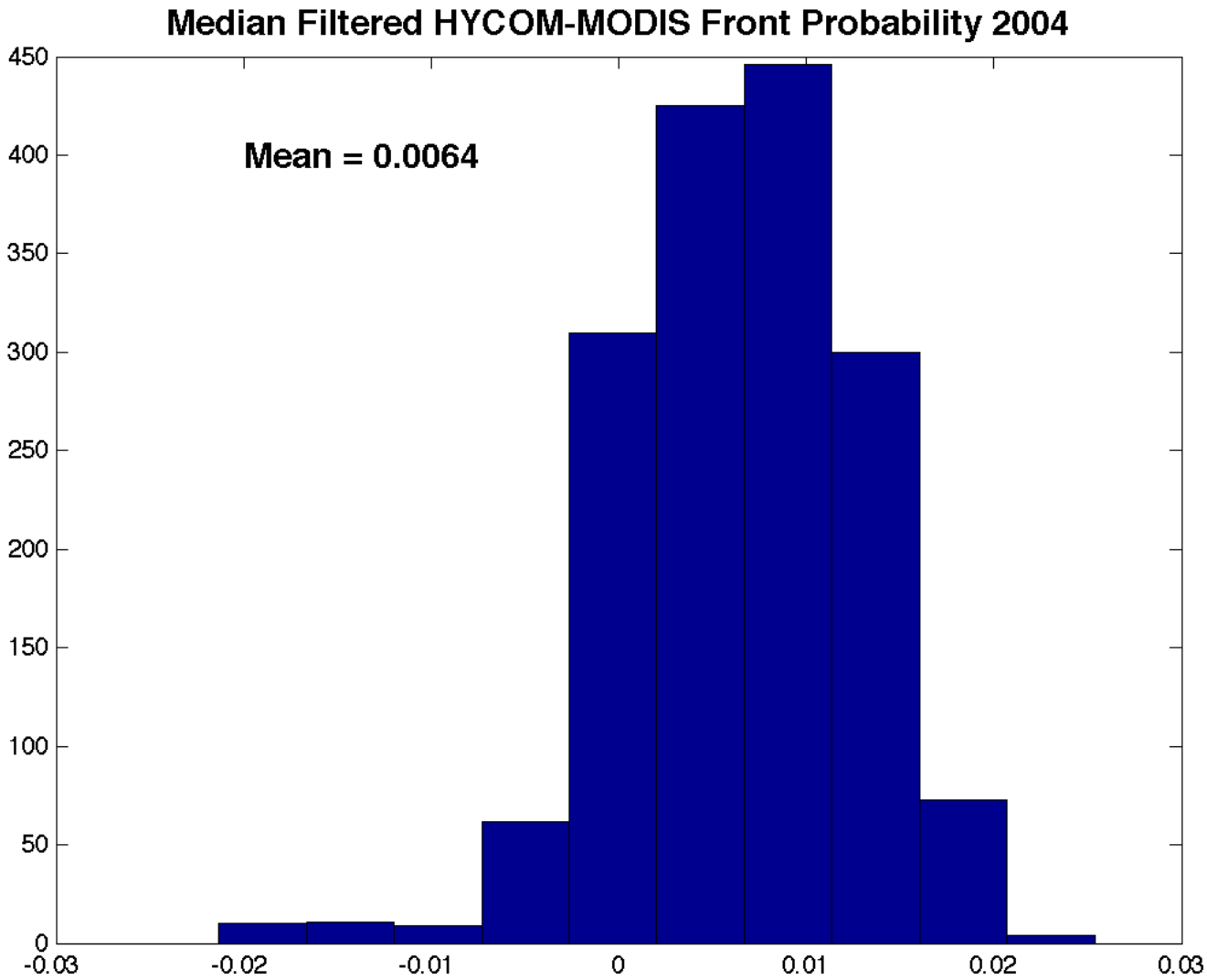


# All Done

The End

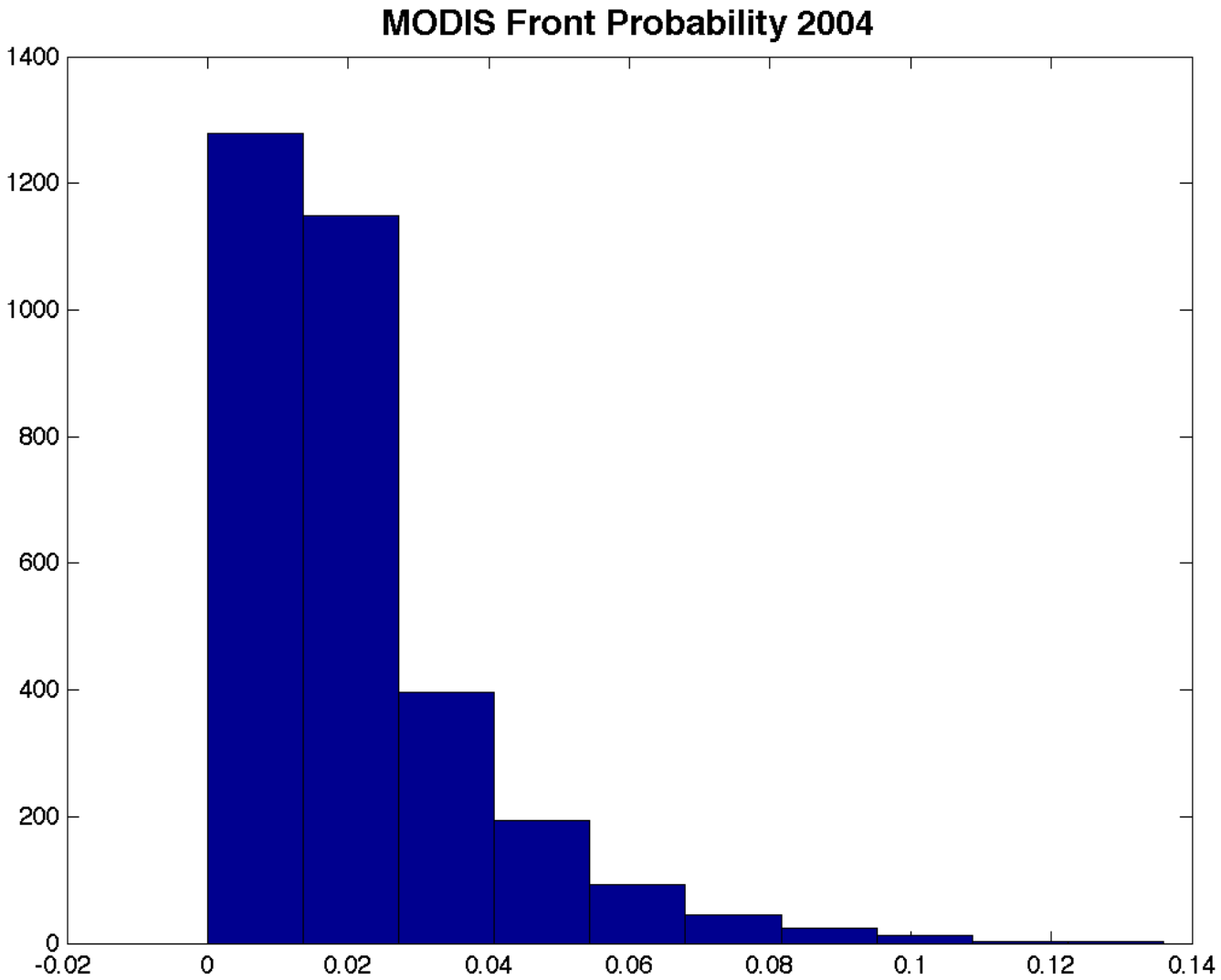
# Median Filtered HYCOM - MODIS Fronl Probability 2004





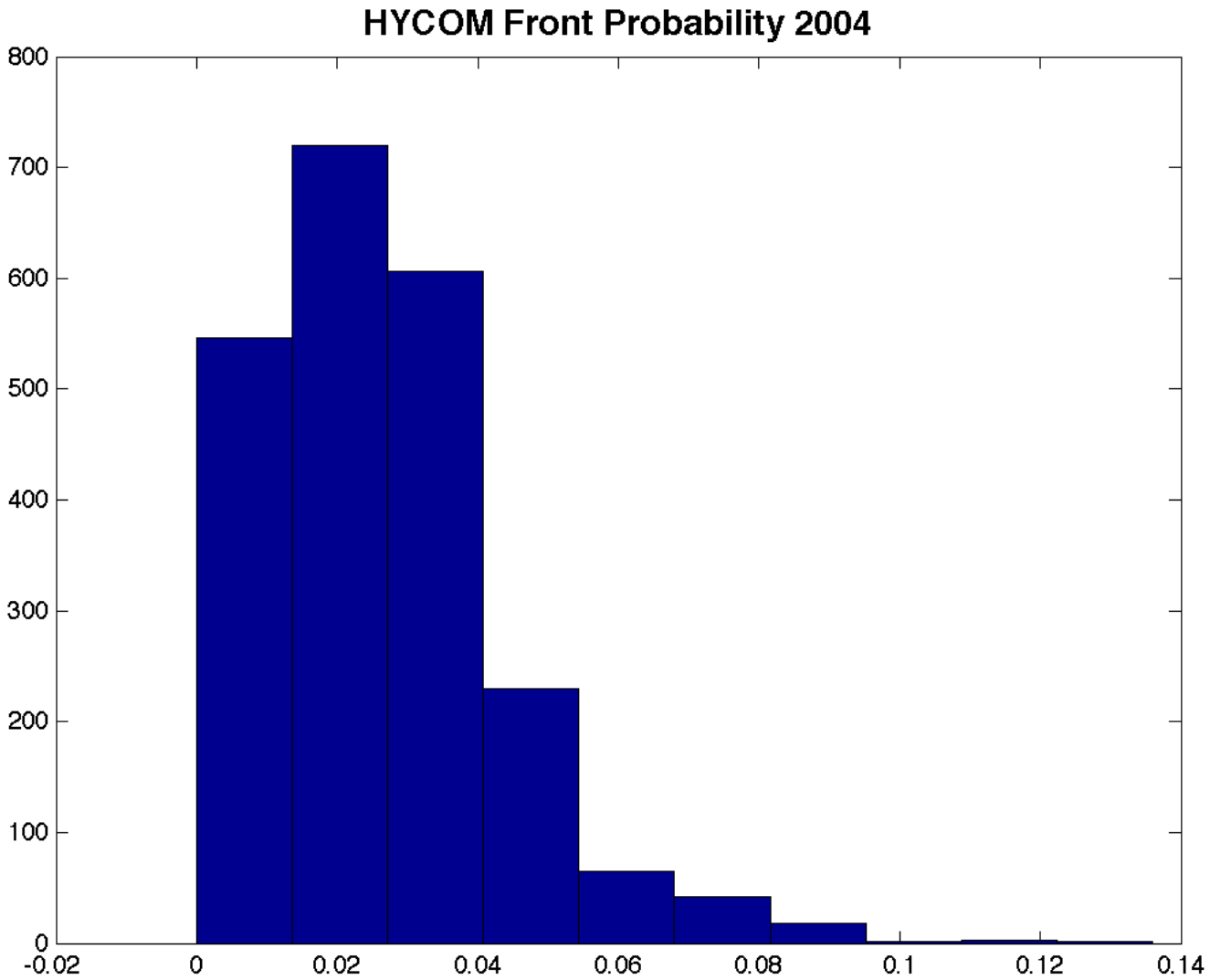
# MODIS Front Probability 2004





# HYCOM Front Probability 2004





# Filtered HYCOM-MODIS Front Probability vs Cloud Cover