**Project Report**
**ATC-275**

# Gust Front Update Algorithm for the Weather Systems Processor (WSP)

S.W. Troxel

29 July 2002

**Lincoln Laboratory**
MASSACHUSETTS INSTITUTE OF TECHNOLOGY
*LEXINGTON, MASSACHUSETTS*

**Prepared for the Federal Aviation Administration.**

**Document is available to the public through the National Technical Information Service, Springfield, Virginia 22161.**

TECHNICAL REPORT STANDARD TITLE PAGE

| | | |
|---|---|---|
| 1. Report No.<br>ATC-275 | 2. Government Accession No. | 3. Recipient's Catalog No. |
| 4. Title and Subtitle<br>Gust Front Update Algorithm for the Weather Systems Processor (WSP) | | 5. Report Date<br>29 July 2002<br>6. Performing Organization Code |
| 7. Author(s)<br>Seth W. Troxel | | 8. Performing Organization Report No.<br>ATC-275 |
| 9. Performing Organization Name and Address<br>MIT Lincoln Laboratory<br>244 Wood Street<br>Lexington, MA 02420-9108 | | 10. Work Unit No. (TRAIS)<br>11. Contract or Grant No.<br>F19628-00-C-0002 |
| 12. Sponsoring Agency Name and Address<br>Department of Transportation<br>Federal Aviation Administration<br>800 Independence Ave., S.W.<br>Washington, D.C. 20591 | | 13. Type of Report and Period Covered<br>Project Report<br>14. Sponsoring Agency Code |

15. Supplementary Notes

This report is based on studies performed at Lincoln Laboratory, a center for research operated by Massachusetts Institute of Technology, under Air Force Contract F19628-00-C-0002.

16. Abstract

The Gust Front Update algorithm (GFUP) is part of the gust front product generation chain for the ASR-9 Weather Systems Processor (WSP). GFUP processes gust front detection and position prediction data output by the Machine Intelligent Gust Front Algorithm (MIGFA), and uses an internal timer to schedule generation of updated current and 10- and 20-minute gust front predictions at 1-minute intervals. By substituting appropriate interval gust front forecast data from MIGFA, the locations of gust fronts shown on the user display are updated at a rate that is faster than the radar base data processed by MIGFA. Prior to output, the updated curve position data are smoothed by GFUP using a tangent-spline interpolation algorithm.

This document provides a general overview and high level description of the GFUP algorithm.

| | | | |
|---|---|---|---|
| 17.Key Words | | 18. Distribution Statement<br>This document is available to the public through the National Technical Information Service, Springfield, VA 22161. | |
| 19. Security Classif. (of this report)<br>Unclassified | 20. Security Classif. (of this page)<br>Unclassified | 21. No. of Pages<br>32 | 22. Price |

FORM DOT F 1700.7 (8-72) Reproduction of completed page authorized

# ABSTRACT

The Gust Front Update algorithm (GFUP) is part of the gust front product generation chain for the ASR-9 Weather Systems Processor (WSP). GFUP processes gust front detection and position prediction data output by the Machine Intelligent Gust Front Algorithm (MIGFA), and uses an internal timer to schedule generation of updated current and 10- and 20-minute gust front predictions at 1-minute intervals. By substituting appropriate interval gust front forecast data from MIGFA, the locations of gust fronts shown on the user display are updated at a rate that is faster than the radar base data processed by MIGFA. Prior to output, the updated curve position data are smoothed by GFUP using a tangent-spline interpolation algorithm.

This document provides a general overview and high level description of the GFUP algorithm.

# TABLE OF CONTENTS

# LIST OF ILLUSTRATIONS

# 1. INTRODUCTION

## 1.1 ALGORITHM PRODUCT DESCRIPTION

**GFUP** is used in the generation of the gust front products. It interprets the output of the Machine Intelligent Gust Front Algorithm (**MIGFA**), containing gust front locations and estimates of future locations along with associated wind shear information, to provide the following:

1. Smoothed symbolic representations of current gust front locations and associated 10- and 20-minute forecast gust front locations.

2. The estimated-time-to-impact (ETI) for the gust front that will first enter a pre-determined gust front impact zone surrounding the airport.

Since sensor measurements that supply input data to **MIGFA** do not provide the desired update rate for gust front outputs and **MIGFA** outputs require additional spatial smoothing, **GFUP** utilizes an internal timer together with forecast gust front location information supplied by **MIGFA** to produce smoothed output at the desired (faster) update rate.

## 1.2 CONCEPTUAL OVERVIEW

For each gust front detection, **MIGFA [1][2][3]** provides associated gust front location forecasts with the desired temporal resolution (via its list of 1-minute interval position forecast data) to allow **GFUP** to update the gust front positions at the required update rate. To do this, **GFUP** utilizes an internal timer as a scheduler. When the timer reaches the next update interval, **GFUP** computes the total elapsed time (this includes any processing latency from **MIGFA** itself as well as the time since the last update) and selects the appropriate forecast curves to represent the "new" current and forecast locations of each gust front.

Gust front outputs provided by **MIGFA** can sometimes contain irregular or jagged curves that are meteorologically unrealistic, as well as being too irregular from a human factors viewpoint, to directly serve as symbolic representations of gust fronts on the user display. It is desirable to produce a smoother approximation of the **MIGFA** output for the end user. In addition, complications can arise when gust fronts collide or split. **MIGFA** can occasionally misinterpret the situation and produce a single set of curve points having a sharp bend at the juncture between the two fronts. Most conventional curve fitting algorithms would have difficulty negotiating such a sharp bend. In fact, the sharp bend represents a natural break point that should not be fitted. The curve should be broken into two segments and smoothing should be applied to the two segments separately. **GFUP** employs a tangent-spline smoothing procedure that is designed to provide an approximating curve for a collection of curve points that have been declared to belong to a gust front. The final product of the tangent-spline procedure is a continuously differentiable spline that approximates the curve points. If it is not possible to create a single curve that provides a good fit of the points, then the final gust front representation may be comprised of two or more tangent-spline segments. Since the curve smoothing can be computationally intensive, it is applied only to those curve points that have been selected to comprise the updated current and 10- and 20-minute forecast locations.

Finally, to satisfy airport planning requirements, **GFUP** computes the estimated-time-to-impact (ETI) for the nearest (in time) gust front in the updated gust front map. By first testing all current gust front locations to see if they intersect the predefined Gust Front Impact Zone, and then expanding the search by 1-minute forecast increments, **GFUP** is able to quickly report the least amount of time until a gust front will intersect the impact zone.

## 1.3 INFORMATION ENVIRONMENT

### 1.3.1 Meteorological Information

Gust front detection and forecast data from the output of the gust front detection algorithm (*MIGFA_gust_fronts*) are the only information required by **GFUP**.

### 1.3.2 Adaptation Parameters

The algorithm processing depends on certain adaptation parameters. It is assumed that the values of these parameters will be made available whenever they are needed for processing. A complete listing of these parameters is provided in 2.3.3.

# 2. HIGH LEVEL ALGORITHM DESCRIPTION

## 2.1 ALGORITHM IDENTIFICATION AND PURPOSE

### 2.1.1 Algorithm Identifier

The algorithm identifier is GFUP.

### 2.1.2 Algorithm Description

**Overview**

**GFUP** is a gust front post-processing algorithm that performs three principal tasks:

1. Create a map of gust front locations and associated 10- and 20-minute forecast locations upon arrival of new input from **MIGFA** or at a prescribed time interval (nominally, one minute) that is smaller than the input data rate.

2. Perform curve smoothing on the output curve points.

3. Compute the estimated-time-to-impact (ETI) of the nearest (in time) gust front with respect to a predefined gust front impact zone surrounding the associated airport.

Figure 1 illustrates data flow through the **GFUP** algorithm. **GFUP** receives its input (*MIGFA_gust_fronts*) from the WSP Machine Intelligent Gust Front Algorithm (**MIGFA**) approximately once every 2 minutes. For each gust front detection, *MIGFA_gust_fronts* contains a set of curve points defining the gust front location, a wind shear hazard number ($\Delta V$), a wind shift estimate, and a wind shift reference point indicating where on the gust front the wind shift analysis was performed (the wind shift reference point is used as the basis for determining the location of the wind shift arrow icon on the display). In addition, **MIGFA** produces a series of curves for each of the detected fronts that provide "snapshots" of future locations of the front at 1-minute intervals out to the **MIGFA** forecast horizon (nominally, 35 minutes). These position forecasts are used by **GFUP** to generate its two outputs: An *updated_gf_map* indicating the current and forecast locations of the gust fronts, and the estimated-time-to-impact (*gf_eti*) of the nearest gust front. Within *updated_gf_map* is a status flag (*gf_update_status*) indicating whether the data in *updated_gf_map* is thought to be reliable and valid for display (*gf_update_status = c_gf_update_ok*), or whether the data is old and unreliable due to an excessive lapse in data received from **MIGFA** (*gf_update_status = c_gf_update_old*). Such a lapse could occur if the radar went down or if there was a problem with **MIGFA** itself.

Figure 2 shows the functional flow of the **GFUP** algorithm. The algorithm starts by waiting for the arrival of gust front detection and forecast data from **MIGFA**. When the next set of **MIGFA** data arrives, the **GFUP_ProcessInit** function ingests all gust front detection and location forecast data and places it in a GF_MAP data structure named *gf_map*. An internal timer is started (*g_update_timer_val* = 0). The initial processing delay due to **MIGFA** processing, *migfa_process_delay,* is computed as the time difference between the radar base data that served as input to **MIGFA** and the current system time.

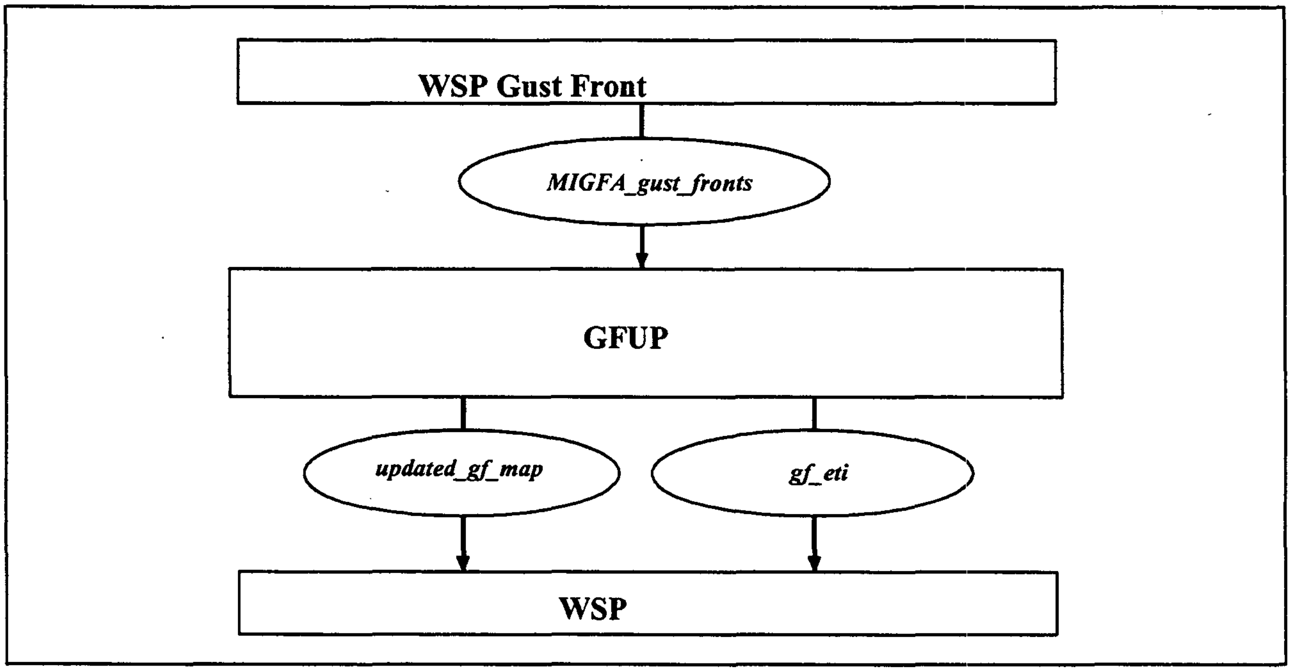


*Figure 1. External interface data flow diagram for the **GFUP** algorithm.*

Next, **GFUP_UpdateLocations** computes the *total_delay* as the sum of the *migfa_process_delay* and the value of the internal timer, *g_update_timer_val.* The *total_delay* is used as an index for selecting and copying 1-minute interval forecast location data into an intermediate GF_MAP (*updated_fine_gf_map*) containing the updated current and 1-minute interval forecast gust front locations (for an example illustration, see Figure 3). The "*fine*" in *updated_fine_gf_map* refers to the 1-minute interval temporal granularity of the forecasts in this map (as compared to the 10-minute temporal granularity in the final output map).

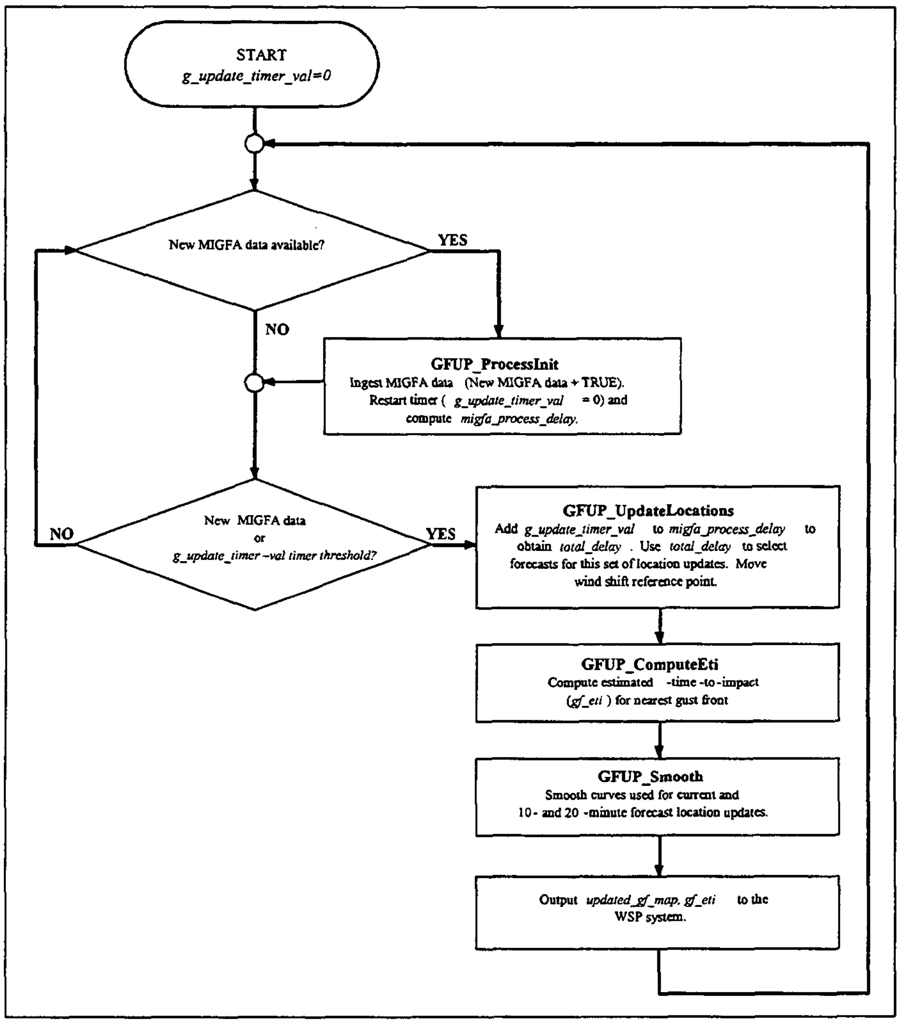


*Figure 2. **GFUP** algorithm functional flow diagram.*

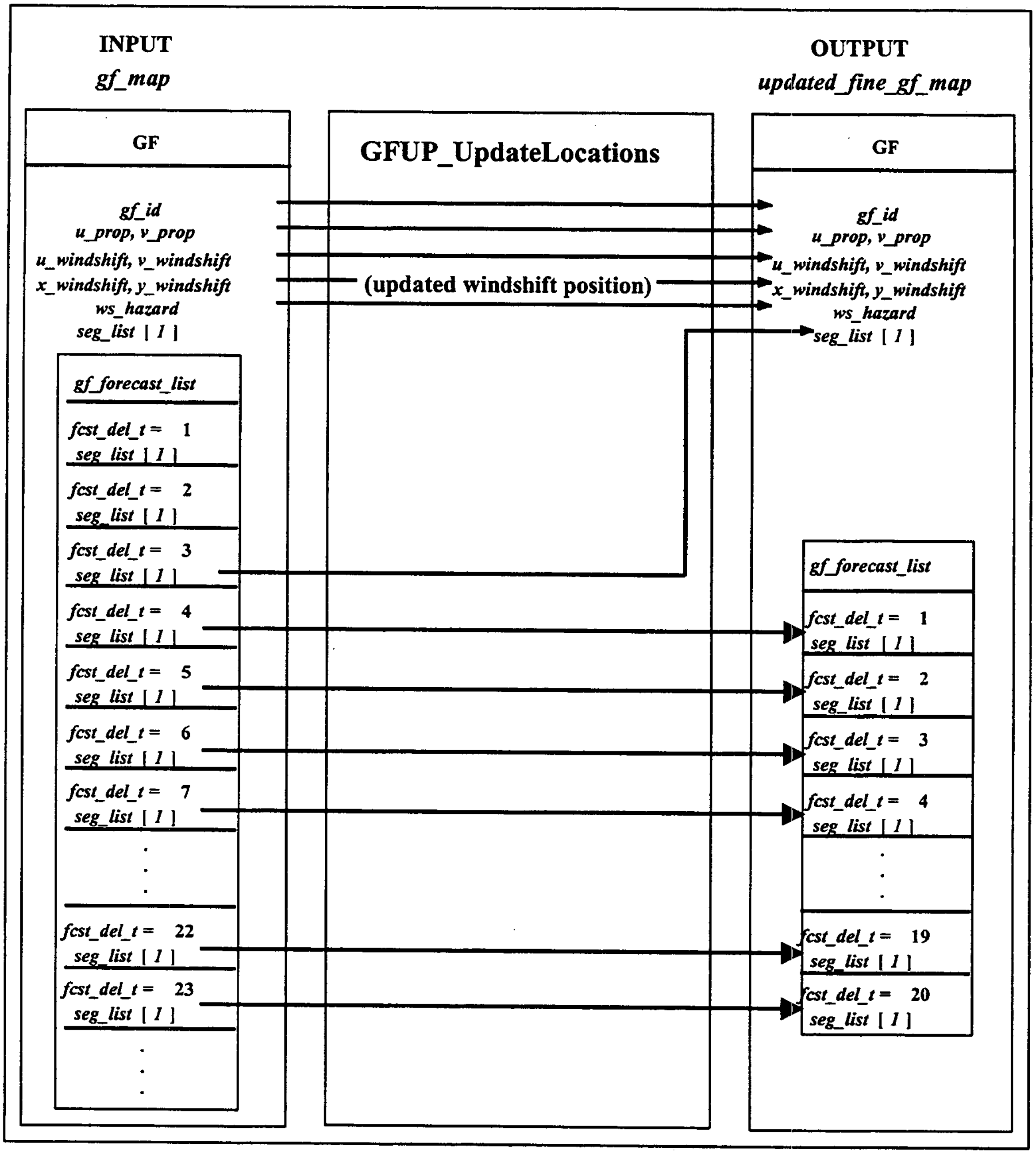


*Figure 3. Data flow diagram for the* ***GFUP_UpdateLocations*** *function showing mapping of the data from a single GF in an input GF_MAP (gf_map) into a corresponding GF in the output GF_MAP (updated_fine_gf_map). For this illustration, a total delay (migfa processing delay + G_update_timer_val) of 3 minutes is assumed. (Not all fields of the GF data structure are shown.)*

The *updated_fine_gf_map* is used by **GFUP_ComputeEti** to determine the estimated-time-to-impact (ETI) to the airport for the nearest (in time) gust front in the GF_MAP. The results are stored in *gf_eti→flag* and *gf_eti→minutes*. *Gf_eti→flag* is a Boolean flag that is set to TRUE if any gust fronts are within range (less than *p_eti_horizon*) or are currently impacting the airport. If the flag is TRUE, then *gf_eti→minutes* contains the number of minutes until gust front impact (zero, if a gust front is currently impacting the airport).

Finally, the *updated_fine_gf_map* is passed to the **GFUP_Smooth** routine. For each gust front in *updated_fine_gf_map*, **GFUP_Smooth** first selects the curve points defining the curves corresponding to the current location and the 10- and 20-minute forecast gust front locations (i.e., 3 curves for each gust front). Each of the selected curves is then smoothed using a tangent-spline technique to replace the curve data points with a new set of smoothed curve data points (more on the tangent-spline procedure shortly). The smoothed gust fronts, along with their 10- and 20-minute forecast locations are placed into a final GF_MAP structure called *updated_gf_map*. The *updated_gf_map* is output along with the ETI information (*gf_eti*) to the WSP system.

Once all of the gust fronts sent by **MIGFA** have been processed to generate the first update, the top-level control loop is reentered. The top-level loop alternately checks for new *MIGFA_gust_fronts* data and checks the value of the timer. If new **MIGFA** data have arrived, the loop is broken and **GFUP_ProcessInit** processes the new **MIGFA** data. Otherwise, if the value of the timer has exceeded the timer threshold established for the next update (based on *p_update_rate*) and no new **MIGFA** data have been received, then a new set of updates (using the forecast location information contained in the last *gf_map*) is generated by passing control directly to **GFUP_UpdateLocations** which adds the value of *g_update_timer_val* to *migfa_process_delay* to obtain a new *total_delay*. The *total_delay* is then used to select and copy another set of forecasts and send a new set of updates to the WSP system. A maximum of *p_max_updates* updates without receiving fresh **MIGFA** data is allowed. After that point, no new maps are computed until new **MIGFA** data is received. Once the maximum number of updates has been exceeded and at each subsequent timer interval, *updated_gf_map"3"gf_update_status* is set to *c_gf_update_old* and the last *updated_gf_map* is resent.

**Tangent-Spline Curve Smoothing Procedure**

Gust front curve data provided by **MIGFA** can contain irregular or jagged curves that are too rough for symbolic representation on the user display. It is desirable to produce a smoother approximation of the curve for the end user. A complication can arise in cases where gust fronts have collided or split. **MIGFA** can misinterpret the situation and produce a single set of curve points that might have a sharp bend at the juncture between the two fronts. Most conventional curve fitting algorithms would have difficulty negotiating such a sharp bend when, in fact, the sharp bend represents a natural break point that should not be fitted. The curve should be broken in two and smoothing should be applied to the two pieces separately.

The tangent spline smoothing procedure invoked by **GFUP_Smooth** is designed to provide an approximating curve for a collection of curve points that have been declared to belong to a gust front. The final product of the tangent-spline procedure is a continuously differentiable spline that approximates the curve points. If it is not possible to create a single curve that provides a good fit within the constraints of the process, then the final gust front curve may be comprised of two or more tangent-spline segments.

There are four major steps to the tangent-spline smoothing procedure (each step is accomplished in the form of a function call):

1. Partitioning (binning) of the input curve points (**Partition**).

2. Tangent construction (**BuildTangents**).

3. Tangent pair analysis (**TangentAnalysis**).

4. Curve tracing (**CurveTrace**).

In the first step (**Partition**), the set of all points is partitioned into subsets based on a rather fine regular grid on the space. The second step (**BuildTangents**) is the computation of tangent lines to these subsets, based on a minimum squared-distance principle. A tangent quality check is used to guarantee that only good tangents are accepted. The third step (**TangentAnalysis**) is a coarse local sieve to determine which tangent pairs could possibly be considered for adjacent members of a curve. The last step (**CurveTrace**) is based on a slope-distance proximity principle. Usually, there is only one reasonable "next point" to add to a curve. In cases where there is more than one reasonable choice, the choice is made based on a test that involves both the distance between the tangent centroids and the difference between the tangent slopes.

The complete function call hierarchy for the **GFUP** algorithm is given in Figure 4.

### 2.1.3 Algorithm Relationship to Other Algorithms

**GFUP** receives gust front detection and forecast data from the **MIGFA** gust front detection algorithm (*MIGFA_gust_fronts*) approximately once every 2 minutes. The outputs of **GFUP** (*updated_gf_map* and *gf_eti*) are sent to the WSP system for alert generation and graphical display.

- **GFUP_Main**
  - **GFUP_ProcessInit**
  - **GFUP_UpdateLocations**
    - **CopyWindInfo**
    - **CopyDetectCurveInfo**
    - **CopyForecasts**
- **GFUP_ComputeEti**
- **GFUP_Smooth**
  - **SmoothCurve**
    - **Partion**
    - **BuildTangents**
      - **CombineSums**
      - **RecordTangent**
      - **Residual**
      - **TangentLength**
    - **Tangent Analysis**
      - **ExtremeDistanceTest**
      - **DistanceSlopeProximityTest**
      - **CurvatureTest**
    - **CurveTrace**
      - **ComputeSearchRegion**
      - **InSearchRegion**
      - **AppendChain**
      - **ChainEnd**
      - **CopyChainToSegment**

*Figure 4. Call tree for **GFUP**.*

### 2.1.4 Algorithm Initialization

Upon startup, *g_update_timer_val* is set to zero.

### 2.1.5 Algorithm Inputs

Input to the **GFUP** algorithm is *MIGFA_gust_fronts* obtained from the output of the WSP gust front detection algorithm (**MIGFA**). Data names and structures provided here are convenient representations of the input data for **GFUP** and do not impose design restrictions outside of the context of the **GFUP** algorithm.

*MIGFA_gust_fronts*

**Description:** Contains all detected gust fronts and associated location forecasts and wind shear estimates from **MIGFA** processing of the latest WSP gust front scan.

**Contents:**

| | |
|---|---|
| *radar_base_time:* | Time of radar base data processed by **MIGFA**. |
| *site_name:* | Radar site identifier. |
| *n_gf_detections:* | Number of GFs in *gf_detection_list.* |
| *gf_detection_list:* | List of GFs (defined below) from *MIGFA_gust_fronts.* |

Each GF in *gf_detection_list* contains the following information:

| | |
|---|---|
| *gf_id:* | Gust front ID number assigned by **MIGFA**. |
| *num_segs:* | Number of CURVE_SEGs (defined below) comprising the curve that represents the current gust front location (*num_segs* = 1 for *MIGFA_gust_fronts*). |
| *seg_list:* | Spatially ordered list of CURVE_SEG structures containing the point lists that represent the current gust front location. |
| *u_prop:* | Eastward component of gust front propagation velocity in m/s. |
| *v_prop:* | Northward component of gust front propagation velocity in m/s. |
| *u_windshift:* | Eastward component of wind velocity behind the front in m/s. |
| *v_windshift:* | Northward component of wind velocity behind the front in m/s. |
| *x_windshift:* | X (East) distance in km from radar to location of wind shift reference point on the gust front. |
| *y_windshift:* | Y (North) distance in km from radar to location of wind shift reference point on the gust front. |

| | |
|---|---|
| *ws_hazard:* | DV (wind shear hazard) in m/s. |
| *n_gf_forecasts:* | Number of 1-minute interval GF_FORECASTs in *gf_forecast_list* (may be zero). |
| *gf_forecast_list:* | List of GF_FORECASTs used for updating current and forecast gust front locations. |

Each GF_FORECAST in *gf_forecast_list* contains the following information:

| | |
|---|---|
| *gf_id:* | Gust front ID number for which this forecast is associated. |
| *fcst_del_t:* | Forecast interval in minutes after initial time of detection. |
| *num_segs:* | Number of curve segments comprising the curve that represents the forecast gust front location (*num_segs* = 1 for *MIGFA_gust_fronts*). |
| *seg_list:* | Spatially ordered list of CURVE_SEG structures containing the point lists that represent the forecast gust front location at time interval *fcst_del_t*. |

A CURVE_SEG is an ordered list of points defining a segment of a gust front curve. Each CURVE_SEG contains the following information:

| | |
|---|---|
| *npts:* | Number of points in the curve segment. |
| *xpts:* | Array of abscissae (x) for the curve segment (km east with respect to radar coordinate system). |
| *ypts:* | Array of corresponding ordinates (y) for the curve segment (km north with respect to radar coordinate system). |

### 2.1.6 Algorithm Outputs

***updated_gf_map***

**Description:** A GF_MAP data structure containing updated current and forecast gust front locations and associated wind shear estimates.

**Contents:**

| | |
|---|---|
| *radar_base_time:* | Time of radar base data processed by **MIGFA**. |
| *reference_time:* | Time for which *updated_gf_map* is representative. |
| *site_name:* | Radar site identifier. |
| *gf_update_status:* | Status indicator for integrity of data in output map. |
| *n_gf_detections:* | Number of GFs in *gf_detection_list*. |

*gf_detection_list:* List of GFs representing updated current and forecast gust front locations and wind information.

Each GF in *gf_detection_list* contains the following information:

*gf_id:* Gust front ID number assigned by **MIGFA**.

*num_segs:* Number of CURVE_SEGs (defined below) comprising the curve that represents the updated current gust front location.

*seg_list:* Spatially ordered list of CURVE_SEG data structures containing the point lists that represent the updated current gust front location.

*u_prop:* Eastward component of gust front propagation velocity in m/s.

*v_prop:* Northward component of gust front propagation velocity in m/s.

*u_windshift:* Eastward component of wind shift behind the front in m/s.

*v_windshift:* Northward component of wind shift behind the front in m/s.

*x_windshift:* X (East) distance in km from radar to location of wind shift reference point on the gust front.

*y_windshift:* Y (North) distance in km from radar to location of wind shift reference point on the gust front.

*ws_hazard:* DV (wind shear hazard) in m/s.

*n_gf_forecasts:* Number of GF_FORECASTs in *gf_forecast_list* (may be zero).

*gf_forecast_list:* List of GF_FORECASTs corresponding to updated forecast gust front locations.

Each GF_FORECAST in *gf_forecast_list* contains the following information:

*gf_id:* Gust front ID number for which this forecast is associated.

*fcst_del_t:* Forecast interval relative to time of updated current gust front location in minutes.

*num_segs:* Number of curve segments comprising the curve that represents the forecast gust front location.

*seg_list:* Spatially ordered list of CURVE_SEG structures containing the point lists that represent the forecast gust front location at time interval *fcst_del_t*.

A CURVE_SEG is an ordered list of points defining a segment of a gust front curve. Each CURVE_SEG contains the following information:

*npts:* Number of points in the curve segment.

*xpts:* Array of abscissae (x) for the curve segment (km east with respect to radar coordinate system).

*ypts:* Array of ordinates (y) for the curve segment (km north with respect to radar coordinate system).

***gf_eti***

**Description:** Estimated-time-to-impact (ETI) in minutes of the nearest (in time) gust front with respect to a predefined gust front impact zone (*p_gfiz*) surrounding the airport.

**Contents:**

*flag:* Boolean status flag whose value is TRUE if there are any gust fronts that are impacting the gust front impact zone (*p_gfiz*) or are forecast to reach the gust front impact zone within *p_eti_horizon* minutes.

*minutes:* Estimated-time-to-impact (ETI) in minutes of the nearest (in time) gust front.

*horizon:* Value of *p_eti_horizon* used as limits for ETI computation.

### 2.1.7 Algorithm Functional Requirements

GFUP shall be able to support internal processing of at least 800 input gust front detection points distributed over as many as 30 gust front curves. Each gust front curve can have as many as 40 associated 1-minute interval gust front forecast curves extrapolated from each point in the original detection curve. Thus GFUP shall be able to support internal processing of at least 32,800 points (800 detection points + 40 forecasts x 800 points). On output, GFUP shall support generation of at least 800 updated gust front detection points distributed over as many as 30 gust front curves. GFUP shall support generation of 2 updated gust front location curves (10- and 20-minute forecasts) for each updated gust front detection curve. Each of the updated forecast curves shall be able to accommodate at least 800 points.

**GFUP** shall produce a maximum of *p_max_updates* consecutive updates between receipts of fresh **MIGFA** data. If *p_max_updates* or more updates have been delivered and new **MIGFA** data have not been received, then the *gf_update_status* flag in the output *updated_gf_map* shall be set to *c_gf_update_old* at each successive timer interval until new **MIGFA** data is received and processed.

## 2.2 ALGORITHM DATA

### 2.2.1 Simple Data Items

| | |
|---|---|
| *g_update_timer_val:* | Timer value in seconds. |
| *migfa_process_delay:* | Delay in seconds due to **MIGFA** processing. |
| *n_active_bins:* | Number of active bins. |
| *n_tan_bins:* | Number of bins with computed tangents. |
| *total_delay:* | Total delay since *gf_map*$^{3}$*radar_base_time* in minutes. |

### 2.2.2 Data Structures

***Instances***

| | |
|---|---|
| *active_bin_info:* | 2-D array of ACTIVE_BIN_INFO structures. |
| *active_bin_list* | List of ACTIVE_BIN_INDEXes. |
| *gf_eti:* | Estimated-time-to-impact (ETI) information for nearest gust front (of type ETI_INFO). |
| *gf_map:* | Input gust front detection and forecast data from **MIGFA** (of type GF_MAP). |
| *tan_bins:* | Array of TAN_BIN_INFO structures. |
| *tan_pairs:* | Array of TAN_PAIR_INFO structures containing tangent pair information. |
| *updated_fine_gf_map:* | Fine scale (1-minute forecast resolution) updated current and forecast gust front locations and associated wind shear estimates (of type GF_MAP). |
| *updated_gf_map:* | Final updated current, and 10- and 20-minute forecast gust front locations and associated wind shear estimates (of type GF_MAP). |

***Structure Types***

ACTIVE_BIN_INDEX:

Contains index data (i, j) denoting the location of an active bin on the partitioning grid. An active bin is one that includes one or more gust front curve points. Each ACTIVE_BIN_INDEX structure contains the following:

*i:* Abscissa of bin location.

*j:* Ordinate of bin location.

ACTIVE_BIN_INFO:

Contains statistics for an active bin in the tangent-spline smoothing algorithm. Each ACTIVE_BIN_INFO structure contains the following:

*Sx:* Summation of xpts(i) values for the active bin.

*Sy:* Summation of ypts(i) values for the active bin.

*Sxx:* Summation of [ xpts(i)*xpts(i) ] values for the active bin.

*Sxy:* Summation of [ xpts(i)*ypts(i) ] values for the active bin.

*Syy:* Summation of [ ypts(i)*ypts(i) ] values for the active bin.

*n_binpts:* Number of input points in the bin.

CHAIN: Collection of Cartesian points for a single chain generated during tangent spline curve fitting. Each CHAIN data structure contains the following:

*pass1_pts:* A list of POINTs containing fitted coordinate points from the first pass of tangent-spline curve tracing.

*pass2_pts:* A list of POINTs containing fitted coordinate points from the second pass of tangent-spline curve tracing.

CURVE_SEG: An ordered list of coordinate points defining a segment of a gust front curve. Each CURVE_SEG structure contains the following:

*npts:* Number of points in the curve segment.

*xpts:* Array of abscissae (x) for the curve segment (km east with respect to radar coordinate system).

*ypts:* Array of ordinates (y) for the curve segment (km north with respect to radar coordinate system).

ETI_INFO: Estimated-time-to-impact (ETI) info for the nearest (in time) gust front. There are two parts to the ETI_INFO data structure:

*flag:* Boolean flag set to TRUE if any gust fronts are within *p_eti_horizon* of the gust front impact zone (*p_gfiz*).

*minutes:* Estimated-time-to-impact for nearest gust front (in minutes).

*horizon:* Value of *p_eti_horizon* used as limits for ETI computation.

GF: Current and forecast gust front location and wind data for a single gust front. A GF structure contains the following information:

*gf_id:* Gust front ID number assigned by **MIGFA**.

*num_segs:* Number of curve segments comprising the curve that represents the current gust front location.

*seg_list:* A spatially ordered list of CURVE_SEG structures containing the point lists that represent the current gust front location.

*u_prop:* Eastward component of gust front propagation velocity in m/s.

*v_prop:* Northward component of gust front propagation velocity in m/s.

*u_windshift:* Eastward component of wind velocity behind the front in m/s.

*v_windshift:* Northward component of wind velocity behind the front in m/s.

*x_windshift:* X (East) distance in km from radar to location of wind shift reference point on the gust front.

*y_windshift:* Y (North) distance in km from radar to location of wind shift reference point on the gust front.

*ws_hazard:* DV (wind shear hazard) in m/s.

*n_gf_forecasts:* Number of 1-minute interval GF_FORECASTs in *gf_forecast_list.*

*gf_forecast_list:* List of GF_FORECASTs used for updating current and forecast gust front locations.

GF_FORECAST: Gust front location forecast data for a single gust front. A GF_FORECAST structure contains the following information:

*gf_id:* Gust front ID number corresponding to the *gf_id* of the associated GF.

*fcst_del_t:* Forecast interval in minutes after initial time of detection.

*num_segs:* Number of curve segments comprising the curve that represents the forecast gust front location.

*seg_list:* A spatially ordered list of CURVE_SEG structures containing the point lists that represent the forecast gust front location.

GF_MAP: Contains all detected gust fronts and associated location forecasts and wind shear estimates from the previous iteration of **MIGFA** processing. A GF_MAP contains the following:

*radar_base_time:* Time of WSP base data processed by **MIGFA**.

*reference_time:* Valid time for which data in GF_MAP is representative.

*site_name:* Radar site identifier.

*gf_update_status:* Status indicator for integrity of data in output map.

*n_gf_detections:* Number of GFs in *gf_detection_list.*

*gf_detection_list:* List of GFs from input *MIGFA_gust_fronts.*

POINT: A simple data structure containing the x,y coordinates of a point in Cartesian radar coordinate space:

*x:* Abscissa of point (km east with respect to radar coordinate system).

y: Ordinate of point (km north with respect to radar coordinate system).

POLYGON: Collection of Cartesian points defining a simple closed polygon.

*num_pts:* Number of points defining the polygon.

*pts:* Array of spatially ordered POINT data structures containing the coordinate points for the polygon.

TAN_BIN_INFO: Contains tangent data for an active bin in the tangent-spline smoothing algorithm. Each TAN_BIN_INFO structure contains the following:

*i, j:* Coordinates of the tangent.

*removed:* Boolean indicating whether point has been removed from further consideration.

*x_centroid, y_centroid:* Centroid location of points in window centered at i, j.

*slope:* Slope of tangent line.

*cosine:* Cosine of the tangent slope angle.

*sine:* Sine of the tangent slope angle.

TAN_PAIR_INFO: Contains analysis data for a pair of tangents. Each TAN_PAIR_INFO structure contains the following:

*dt:* Slope difference between the two tangents.

*l:* Along-tangent distance between the two tangents.

*w:* Cross-tangent distance between the two tangents.

*match:* Match category for the two tangents.

TS_POINT_INFO: Contains information for mapping an original curve point (x,y) to its partitioning grid location (i, j). Each TS_POINT_INFO structure contains the following:

*i:* Abscissa of bin location in the partitioning grid.

*j:* Ordinate of bin location in the partition grid.

*x:* Abscissa of point.

y: Ordinate of point.

### 2.2.3 Algorithm Parameters

| | |
|---|---|
| **NOTES:** | Parameters with a "*p_ts_*" prefix are parameters associated with the tangent-spline curve smoothing procedure. |
| *p_gfiz:* | Ordered list of Cartesian x,y points (km east and north with respect to the radar coordinate system) of type POLYGON forming a simple closed polygon that defines a gust front impact zone around an airport for purposes of generating estimated-time-to-impact reports. |
| *p_max_updates:* | Maximum number of update cycles allowed between receipts of fresh gust front detection data from **MIGFA**. |
| *p_ts_bin_size:* | Size of square partitioning bins (in X or Y direction) in kilometers. |
| *p_ts_centroid_tol:* | Maximum allowable distance of a point from the centroid. |
| *p_ts_correlation_tol:* | Minimum correlation coefficient for a tangent. |
| *p_ts_curve_resolution:* | Resolution factor for controlling the output curve point density. |
| *p_ts_instability_ratio:* | Maximum acceptable ratio of fitted curve length to straight-line distance between two successive gust front points. |
| *p_ts_merge_length:* | Distance within which tangents will be merged (category 8). |
| *p_ts_length_1:* | Maximum along-tangent distance for category 1. |
| *p_ts_length_2:* | Maximum along-tangent distance for category 2. |
| *p_ts_max_slope:* | Maximum slope value (vertical tangent). |
| *p_ts_min_match_len:* | Minimum distance between associated tangents. |
| *p_ts_min_tan_pts:* | Minimum # of points in window needed for tangent computation. |
| *p_ts_n_bins:* | Extent of region (in bins) from the base point in x or y direction. |
| *p_ts_spread_tol:* | Maximum spread for a tangent. |
| *p_ts_tangent_1:* | Slope difference for category 1. |
| *p_ts_tangent_2:* | Slope difference for category 2. |
| *p_ts_merge_slope_diff:* | Minimum slope difference threshold for merging similar tangents. |
| *p_ts_variance_tol:* | Minimum variance for a horizontal or vertical tangent. |
| *p_ts_win_half_width:* | Window half-width (integer). |
| *p_ts_width_1:* | Maximum cross-tangent distance for category 1. |
| *p_ts_width_2:* | Maximum cross-tangent distance for category 2. |
| *p_ts_x_high:* | Abscissa of upper right corner of partitioning grid. |

| | |
|---|---|
| *p_ts_x_low:* | Abscissa of lower left corner of partitioning grid. |
| *p_ts_y_high:* | Ordinate of upper right corner of partitioning grid. |
| *p_ts_y_low:* | Ordinate of lower left corner of partitioning grid. |
| *p_update_rate:* | Rate (in seconds) at which updated gust front positions are computed and sent to the WSP system. |

# APPENDIX A
# PARAMETER TABLE FOR GUST FRONT UPDATE (GFUP)

| Parameter Name | Nominal Value | Units | Range | Precision |
|---|---|---|---|---|
| *P_eti_horizon:* | 20 | minutes | 0 to 20 | 1 |
| *P_gfiz:* ** | | | | |
| *num_pts:* | 4 | unitless | 0 to 100 | 1 |
| *pts [1]:* | | | | |
| *x:* | -5.0 | km | -100.0 to 100.0 | 0.1 |
| *y:* | 15.4 | km | -100.0 to 100.0 | 0.1 |
| *pts [2]:* | | | | |
| *x:* | 7.0 | km | -100.0 to 100.0 | 0.1 |
| *y:* | 15.4 | km | -100.0 to 100.0 | 0.1 |
| *pts [3]:* | | | | |
| *x:* | 7.0 | km | -100.0 to 100.0 | 0.1 |
| *y:* | -15.4 | km | -100.0 to 100.0 | 0.1 |
| *pts [4]:* | | | | |
| *x:* | -5.0 | km | -100.0 to 100.0 | 0.1 |
| *y:* | -15.4 | km | -100.0 to 100.0 | 0.1 |
| *p_max_updates:* | 7 | unitless | 0.0 to 100.0 | 1 |
| *p_ts_bin_size:* | 2 | km | 0.0 to 10.0 | 1 |
| *p_ts_centroid_tol:* | 3.0 | km | 0.0 to 10.0 | 0.1 |
| *p_ts_correlation_tol:* | 0.1 | unitless | 0.0 to 1.0 | 0.1 |
| *p_ts_curve_resolution:* | 2.0 | unitless | 0.0 to 10.0 | 0.1 |
| *p_ts_instability_ratio:* | 1.25 | unitless | 0.0 to 1000.0 | 0.01 |
| *p_ts_merge_length:* | 1.0 | km | 0.0 to 100.0 | 0.1 |
| *p_ts_length_1:* | 5.0 | km | 0.0 to 100.0 | 0.1 |
| *p_ts_length_2:* | 10.0 | km | 0.0 to 100.0 | 0.1 |
| ** Each instance of the algorithm for a given site may require a unique value of this parameter. | | | | |

| Parameter Name | Nominal Value | Units | Range | Precision |
|---|---|---|---|---|
| *p_ts_max_slope:* | 1000.0 | unitless | 0.0 to 1000.0 | 1 |
| *p_ts_min_match_len:* | 1.0 | km | 0.0 to 100.0 | 0.1 |
| *p_ts_min_tan_pts:* | 3 | unitless | 0.0 to 100.0 | 1 |
| *p_ts_n_bins:* | 140 | unitless | 0.0 to 200.0 | 1 |
| *p_ts_spread_tol:* | 0.3 | unitless | 0.0 to 1.0 | 0.1 |
| *p_ts_tangent_1:* | 0.5 | unitless | 0.0 to 1000.0 | 0.1 |
| *p_ts_tangent_2:* | 5.0 | unitless | 0.0 to 1000.0 | 0.1 |
| *p_ts_merge_slope_diff:* | 1.0 | unitless | 0.0 to 1000.0 | 0.1 |
| *p_ts_variance_tol:* | 0.01 | km | 0.0 to 1.0 | 0.01 |
| *p_ts_win_half_width:* | 2 | unitless | 0.0 to 10.0 | 1 |
| *p_ts_width_1:* | 5.0 | km | 0.0 to 100.0 | 0.1 |
| *p_ts_width_2:* | 10.0 | km | 0.0 to 100.0 | 0.1 |
| *p_ts_x_high:* | 90 | km | -100.0 to 100.0 | 1 |
| *p_ts_x_low:* | -90 | km | -100.0 to 100.0 | 1 |
| *p_ts_y_high:* | 90 | km | -100.0 to 100.0 | 1 |
| *p_ts_y_low:* | -90 | km | -100.0 to 100.0 | 1 |
| *p_update_rate:* | 60 | sec | 0.0 to 1000.0 | 1 |

# GLOSSARY

| | |
|---|---|
| ETI | Estimated-time-to-impact |
| GFUP | Gust Front Update algorithm |
| MIGFA | Machine Intelligent Gust Front Algorithm |
| WSP | Weather Systems Processor |

# REFERENCES

1. R. L. Delanoy and S. W. Troxel, 1993: "Automated gust front detection using knowledge-based signal processing," IEEE National Radar Conference, Boston, MA.

2. S. W. Troxel and R. L. Delanoy, 1994: "Machine intelligent approach to automated gust front detection for Doppler weather radars," SPIE Vole. 2220, Image Sensing, Processing, and Understanding for Control and Guidance for Aerospace Vehicles, Orlando, FL.

3. S. W. Troxel and W. L. Pughe, "Machine Intelligent Gust Front Algorithm for the WSP," MIT Lincoln Laboratory Project Report ATC-274, June 2002.