

Computational Method to Calculate Flow Rate Distribution through Radiator

Thomas N. Ramsay*

ABSTRACT

In this paper a computational method to calculate the flow rate distribution through a vehicle radiator is outlined. A typical testing method for radiator flow rate distribution is to cover the front grille openings in various combinations and then measure - via anemometers - the change in radiator flow rate to determine the contribution of a specific cooling opening to the overall cooling flow. However, this changes the contribution distribution: a low pressure region will be created behind the covered grille and will draw air from another area. It would be desirable to capture the contribution air flow distribution through the radiator during normal operating modes (e.g. when all the grille openings are clear). This would be challenging in a testing scenario, but relatively straightforward for a computational fluid dynamics (CFD) analysis. Rather than using the typical velocity magnitude or velocity vector visualization techniques, tracing path lines downstream from the cooling openings to the radiator face leads to a method to allow a more realistic description as well as a better way to convey this information. Basically, the intersection points of the path lines are trapped on the face of the radiator, and then triangulated, and the flow distribution is calculated using these triangular elements. Finally, an image is created to visually convey the flow rate distribution.

1. Introduction

In a typical testing scenario, during an engine airflow test, the front grilles would be covered in various combinations and then measure - via anemometers - the change in radiator flow rate to determine the contribution of a specific cooling opening to the overall cooling flow at the front of the radiator. However, this testing scenario does not capture the on-road contribution distribution from each cooling zone: a low pressure region will be created behind a closed grille and will simply be “filled in” by the flow field from another area - thereby potentially creating an unrealistic contribution distribution. A desirable method would be to capture the contribution air flow distribution through the radiator during normal operating modes (e.g. when all the grille openings are clear). This would be challenging in a testing scenario, but relatively straightforward for a CFD analysis. By realizing that path lines will allow downstream tracing of the flow field from the grille openings to the radiator face, a method was developed to allow a more realistic description as well as an imaging technique to convey this information. This method was developed in-house and contains four main steps: (1) path line intersection with radiator face, (2)

triangulation of intersection points, (3) calculation of individual flow rates and edge export, and (4) image creation. It is the concept of employing a triangulation algorithm (Delauney triangulation in this case) to connect the path line intersection points that makes this method novel. To the author’s knowledge, this is the first time this particular technique has been used to make a useful link between the styling-related grille openings and the performance-related cooling flow through the radiator.

2. Path Line Intersection

A full engine compartment CFD model was created and solved at typical US highway speeds, as seen in Figure 1, where the left side identifies the cooling regions and the left side shows the path lines into these cooling regions. Industry standard CFD setup parameters and solution convergence criteria were used.

Figure 2 shows the route the path lines progress from the cooling openings (grilles or bottom breather) to the radiator face. Note that the four colored areas - blue, yellow, green, and pink - represent the nexus of points that the path lines were released from. The large open arrows indicate the general direction of the cooling flow path lines from associated cooling region toward

* HONDA R&D AMERICAS, INC.

the radiator. Keep in mind that the background colors in Figure 2 will be used throughout the rest of the paper to help differentiate the originating cooling region (blue = upper grille, yellow = lower left grille, green = lower right grille, pink = bottom breather).

Path lines were generated and saved - including the coordinates and the velocity magnitudes - for all four engine cooling zones, the upper grille, the bottom breather or air dam area under the vehicle, and the lower

left and right openings. Figure 3 shows the path lines created downstream of the four individual cooling zones.

Two custom scripts were written to read in the path line file, intersect those path lines with the radiator face and then to trap all the intersection points that were within the rectangular radiator face, since some path lines would bypass the radiator altogether. Figure 4 shows these intersection points colored by originating cooling zone (blue = upper grille, pink = bottom

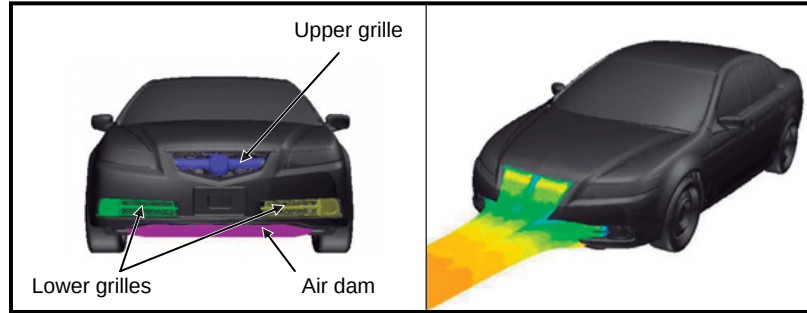


Figure 1 Front end geometry (left) and path lines passing through cooling zones

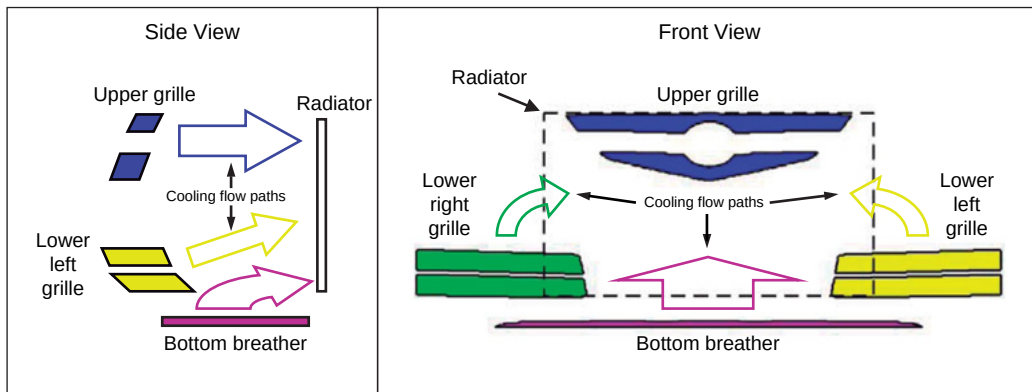


Figure 2 Conceptual path line image

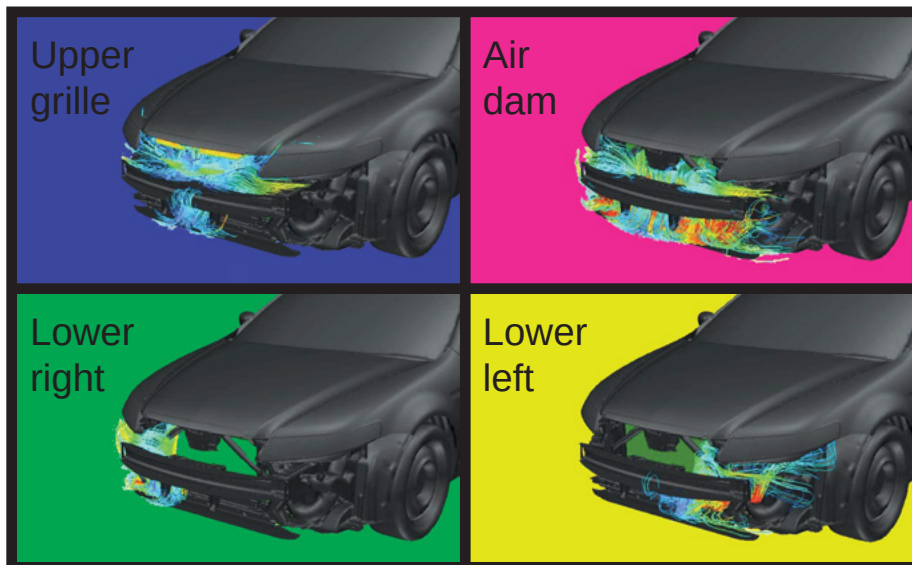


Figure 3 Path line creation

breather, yellow = lower left grille, and green = lower right grille). Each point represents the intersection of a single path line and the front radiator face.

Upon a close inspection of Figure 4, it is easily seen that there are several intersection points that are considered outliers (e.g. the blue points in the yellow region or the pink points in the green region). These points would then be determined to be outside the core contribution region for that cooling zone. Also note that due to a more direct path, the upper grille intersection points tended to have a higher density than the other three regions; however, this is also a function of the originating number of points, which is dependent on the cooling region area and mesh size.

3. Point Triangulation

Now that the relevant points on the front face of the radiator have been found, those points must be associated in some way to use them as a basis for flow rate calculation. The best way to do this is Delauney triangulation, which basically connects all the points with a series of triangular elements, in effect meshing the region, but using the pre-existing points as nodes. Figure 5 shows the Delauney triangulation of seven points, where a triangular element is created from the first three points, then as each subsequent point is added, the points are re-triangulated for a best fit of elements. This is accomplished by use of circumscribed circles. If the

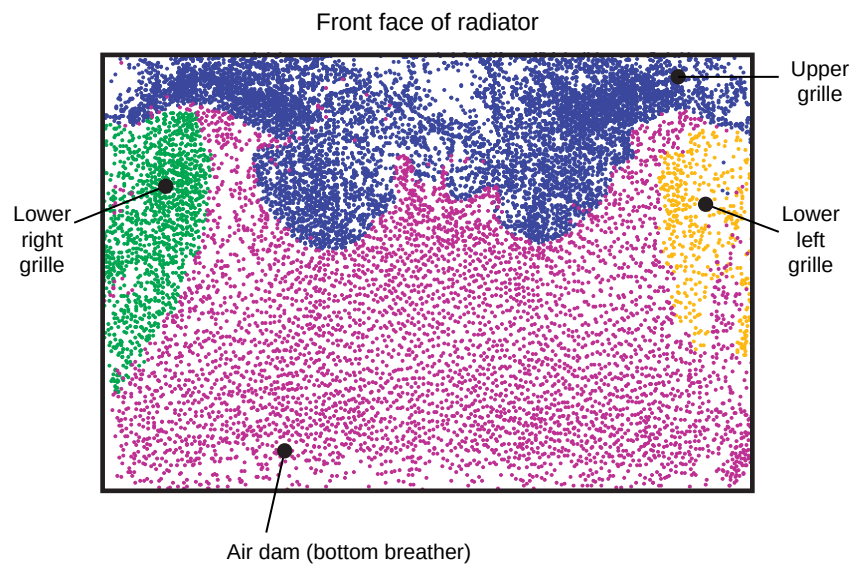


Figure 4 Path line intersection for four radiator cooling areas

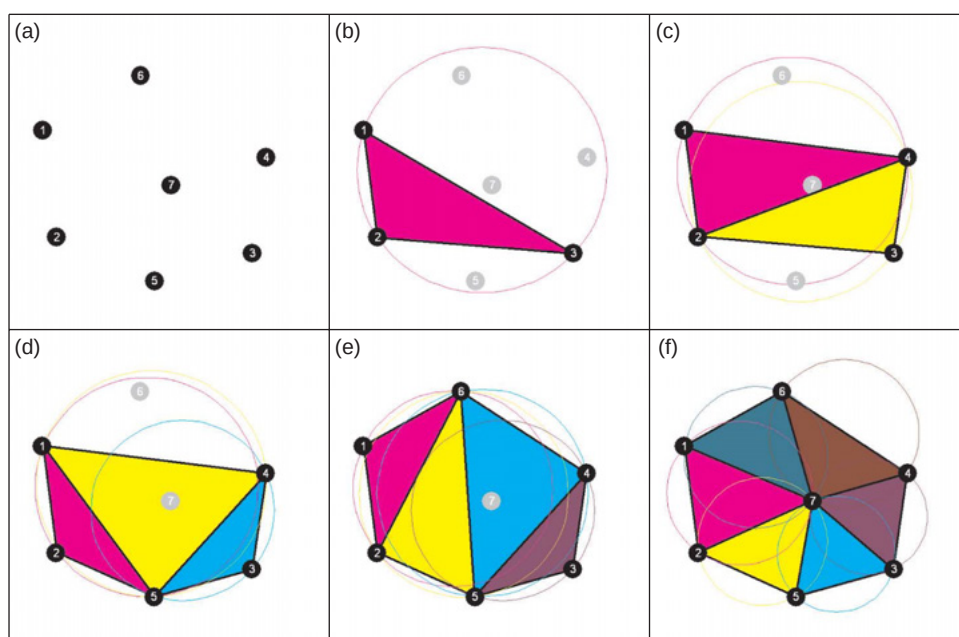


Figure 5 Delauney triangulation, forward progression

next point to be added, such as point #4 in Figure 5(b) is within any of the circumscribed circles represented by an element, then those elements will be re-triangulated with the new point. This is best shown in Figure 5(d), where the point #6 is within the circles of the pink and yellow elements, and thus re-triangulated in Figure 5(e) with the new pink, yellow, and light blue elements. Figure 6 shows the same seven points, but with the addition of triangulated points in reverse order. While the individual triangular elements are created in a different manner, the end result is the same: six triangular elements that contain no grid points within them.

Once the intersection points are triangulated, edge elements are created. In the case of Figure 5 and Figure 6, there are six edge elements (1-2, 2-3, 3-4, 4-5, 5-6, 6-1). These are found by tracking how many elements are associated with each point. Those points with only one or two elements are considered edge points. These edge points are then connected with 1D elements and those elements are sorted to form a closed

edge profile.

A realistic set of intersection points will not form a well shaped series of triangular elements, so a recursive algorithm that removes highly skewed edge elements was added. Returning to the point set of Figure 5 and Figure 6, Figure 7 shows what happens when point #6 is moved and another element is created (in red). It's pretty clear that the new (red) edge triangular element is ill-formed and would not realistically contribute to the flow rate calculation, thus would be eliminated, as in Figure 7(c).

The typical criterion for ill-formed elements is called skew and is defined based on the following equation:

$$Skew = \frac{\text{Element Area}}{\text{Ideal Area}} \quad (1)$$

Where the element area is simply the area of the triangular element, as in Figure 8(a) and the ideal element is the area of a perfect isosceles triangle that lies within the circumscribed circle of that same element, as in Figure 8(c). The skew limit was set at 0.80.

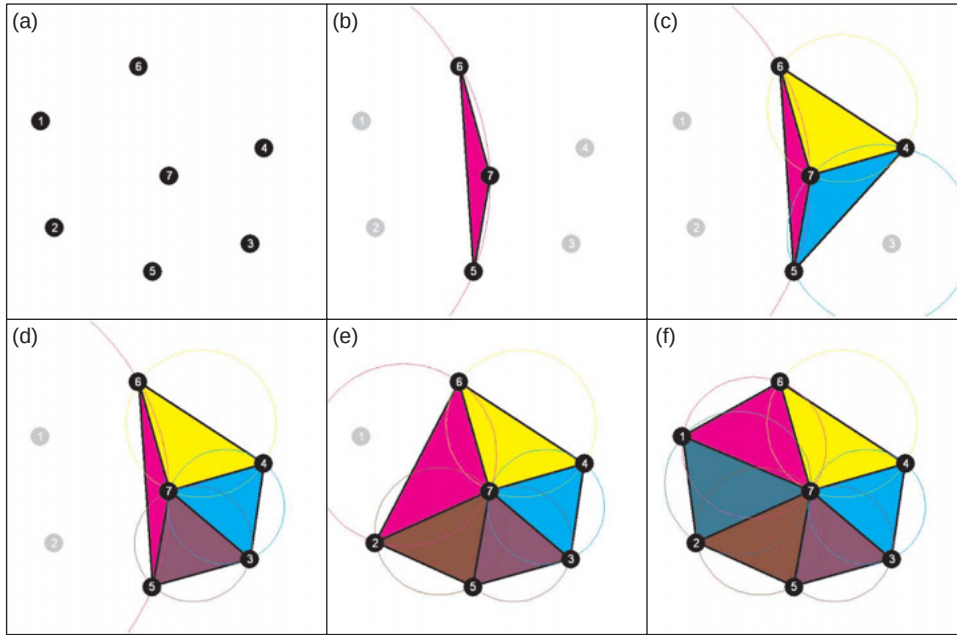


Figure 6 Delaunay triangulation, reverse progression

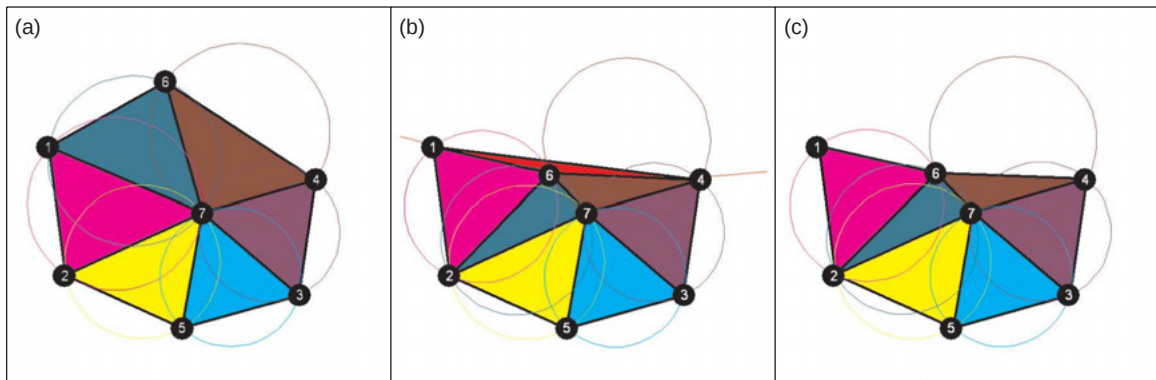


Figure 7 Triangulation for a moved point (#6)

Additionally, it was also possible that larger concave sections would be present in the point set and then large bridging elements would be created. While most of these elements would be likely eliminated by the edge-skew check, it is sometime necessary to indicate that these bridging elements are again outside the core contribution region for that particular cooling zone. Figure 9 shows the triangulated elements for the four cooling zones on the front face of the radiator. Note that there is some overlap in the regions.

4. Flow Rate Calculation

Now that the individual contribution regions have been triangulated, their flow rates can be calculated. Here is where the wisdom of carrying the velocity magnitude for each point comes into play. Realizing that the flow rate of each element can be calculated by:

$$\text{Flow Rate} = \text{Element Area} \times \text{Average Velocity} \quad (2)$$

Where the average velocity is simply computed from the velocity at all three node points. Then all the flow

rates are summed up and the total flow rate through the region of interest is computed. Figure 10 shows the total flow rates (in percentages) for the four contribution cooling zones. In addition to the contribution flow rates, the edge elements are exported as well.

5. Image Creation

Now that the contribution flow rates are calculated, this information is sent back to engineering and styling. While bar graphs of the contributions are relatively straightforward and may be created for engineering, styling prefers more visual results. Therefore, the edge elements from Figure 10 are then plotted and filled in via a simple computer drawing package. The final image is shown in Figure 11 and represents a typical contribution image that would then be shown in an analysis of the flow field. By typical image, it has been shown through past experimental testing that the bottom breather contributes a significant amount of cooling flow, but not how much (for reasons stated in the Introduction). Thus, Figure 11 represents what would be expected from a bottom breather vehicle.

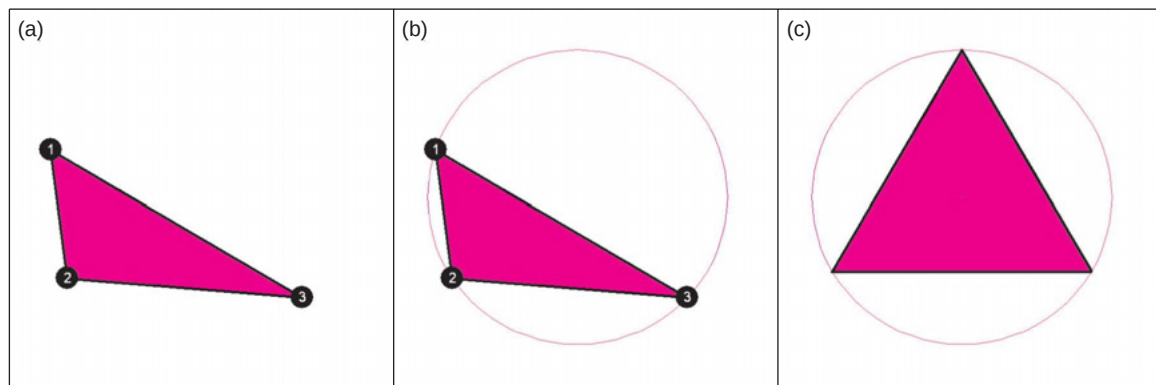


Figure 8 Element skew calculation

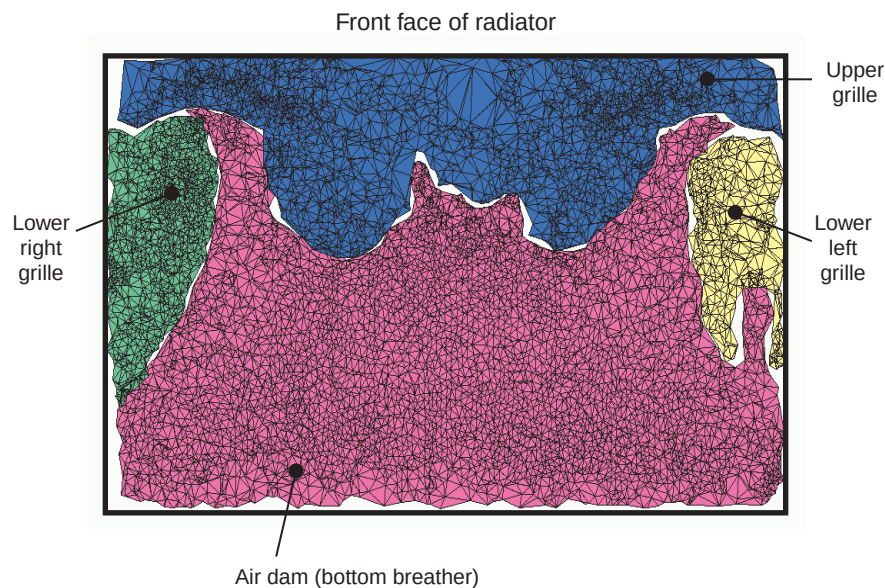


Figure 9 Triangulated contribution regions for candidate radiator

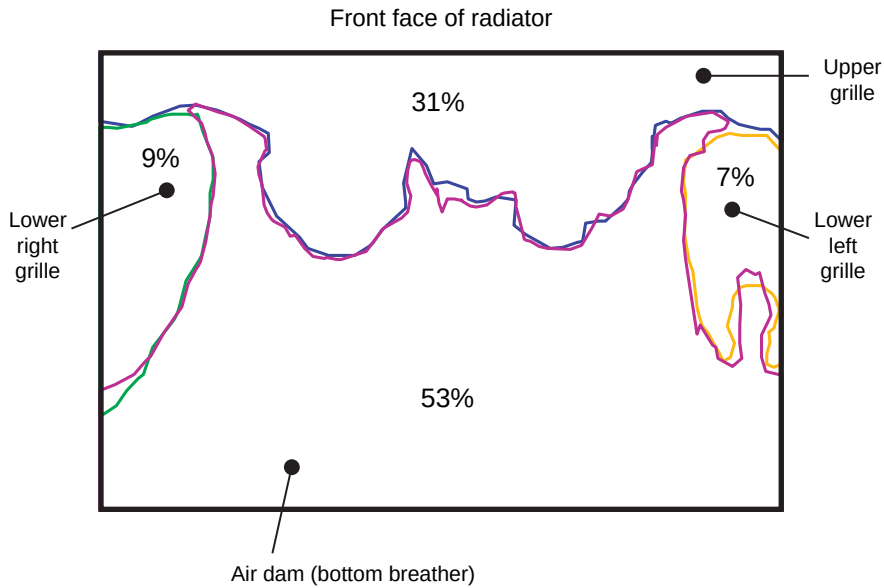


Figure 10 Radiator flow rates for the cooling zones

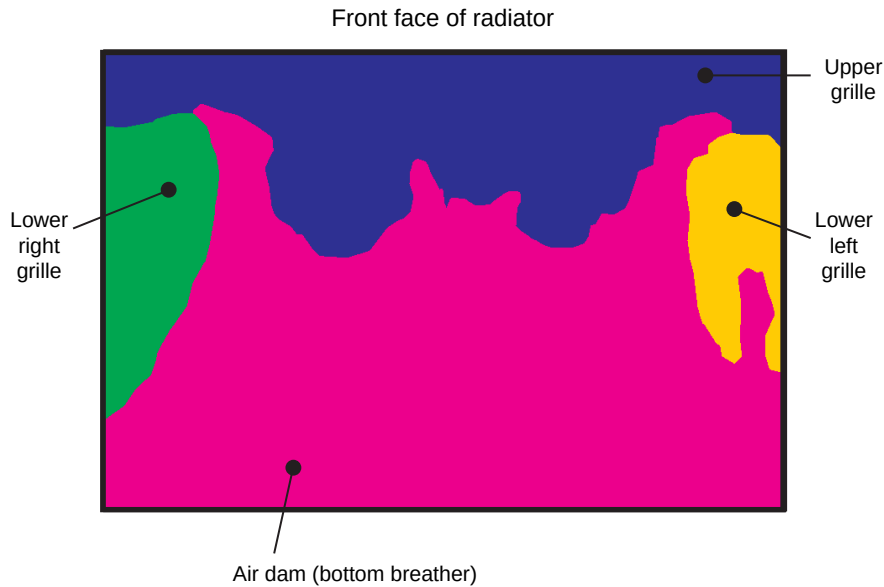


Figure 11 Contribution region final image

6. Conclusions

A method to both calculate and show the cooling zone distribution on the radiator face has been developed. In developing the proposed method, the initial impetus was to convey complex underhood flow patterns. This was accomplished by connecting CFD results (path lines) with a desired outcome (radiator flow distribution). While at this time it is quite difficult to verify this final image (e.g. Figure 11), this method is another tool that allows greater understanding of the cooling region flow field. This proposed method allows enhanced potential front end design direction to be delivered to engineers and stylists alike and increases the confidence in CFD to deliver unique flow field related

results. This better insight into the affects of front fascia design on the flow through the radiator, would, in turn, lead to better styling and performance decisions.

■著 者■



Thomas N. Ramsay