BỘ GIÁO DỤC VÀ ĐÀO TẠO

**TRƯỜNG ĐẠI HỌC CẦN THƠ**

**BÁO CÁO TỔNG KẾT**

**ĐỀ TÀI NGHIÊN CỨU KHOA HỌC CỦA SINH VIÊN**

**ỨNG DỤNG MẠNG CẢM BIẾN XÂY DỰNG HỆ THỐNG GIÁM SÁT NHIỆT ĐỘ KHO LẠNH TRONG KHÔNG GIAN BA CHIỀU**

**TSV2020-50**

**Cần Thơ, 11/2020**

BỘ GIÁO DỤC VÀ ĐÀO TẠO

**TRƯỜNG ĐẠI HỌC CẦN THƠ**

**BÁO CÁO TỔNG KẾT**

**ĐỀ TÀI NGHIÊN CỨU KHOA HỌC CỦA SINH VIÊN**

**ỨNG DỤNG MẠNG CẢM BIẾN XÂY DỰNG HỆ THỐNG GIÁM SÁT NHIỆT ĐỘ KHO LẠNH TRONG KHÔNG GIAN BA CHIỀU**

**TSV2020-50**

Sinh viên thực hiện: **Trần Vi Khan** Nam, Nữ: **Nam**

Dân tộc: **Kinh**

Lớp, khoa: **KTPM 1, Khoa CNTT&TT**

Năm thứ: **4**/Số năm đào tạo: **4.5**

Ngành học: **Kỹ thuật phần mềm**

Người hướng dẫn: **TS. Trương Minh Thái**

**Cần Thơ, 11/2020**

**CÁC THÀNH VIÊN THAM GIA NGHIÊN CỨU**

**VÀ ĐƠN VỊ PHỐI HỢP CHÍNH**

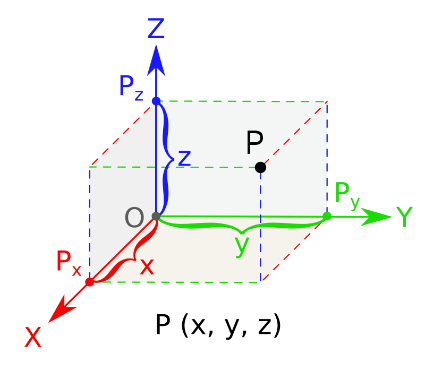
1. **THÀNH VIÊN THAM GIA NGHIÊN CỨU ĐỀ TÀI**

|  |  |  |  |
| --- | --- | --- | --- |
| **TT** | **Họ và tên** | **Vai trò** | **MSSV/MSCB, Đơn vị** |
| 1 | Trương Minh Thái | Cán bộ hướng dẫn | 00520  Khoa CNTT&TT |
| 2 | Trần Vi Khan | Chủ nhiệm đề tài | B1704736  Lớp KTPM 1, K43 |
| 3 | Dương Ý Nguyện | Thành viên chính | B1704759  Lớp KTPM 1, K43 |
| 4 | Nguyễn Quốc Toàn | Thành viên chính | B1704780  Lớp KTPM 1, K43 |

1. **ĐƠN VỊ PHỐI HỢP CHÍNH**

|  |  |  |
| --- | --- | --- |
| **Tên đơn vị** | **Nội dung phối hợp** | **Họ và tên người đại diện**  **đơn vị** |
| Bộ môn Công Nghệ  Thực Phẩm khoa Nông  Nghiệp, Đại học Cần  Thơ | Nghiên cứu mô hình quản lý nhiệt độ trong kho lạnh.  Thực nghiệm đề tài nghiên cứu. | PGS.TS Nguyễn Công Hà  (Trưởng bộ môn ) |

CHƯƠNG 1: MÔ HÌNH HÓA PHÂN BỐ NHIỆT ĐỘ KHO LẠNH BẰNG PHƯƠNG PHÁP NỘI SUY BA CHIỀU VÀ THƯ VIỆN ĐỒ HỌA 3D THREE JS

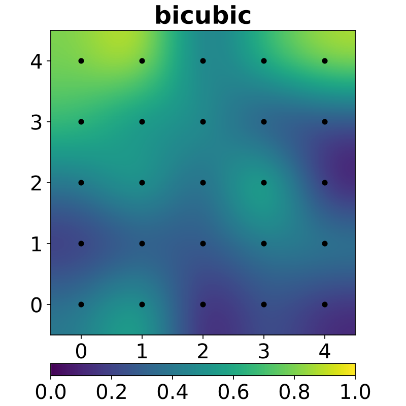
1. PHƯƠNG PHÁP NỘI SUY BA CHIỀU
2. Cơ sở lý thuyết
   1. Không gian ba chiều

Không gian ba chiều là một mô hình hình học có ba thông số, trong đó bao gồm tất cả các vật chất được chúng ta biết đến. Ba chiều được nhắc đến ở đây thường là chiều dài (x), chiều rộng (y), chiều cao (z). Ba hướng bất kì nào cũng có thể được chọn, miễn là chúng không nằm trong cùng một mặt phẳng.

Hình 1: Minh họa không gian ba chiều

* 1. Nội suy đa biến (Multivariate interpolation)

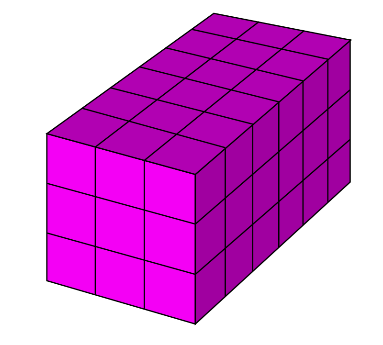
Nội suy đa biến là phương pháp nội suy với các hàm có nhiều hơn một biến. Hàm nội suy trên các điểm đã cho (a, b, c, …) với giá trị đầu ra ở các điểm tùy ý cần xét. Nội suy đa biến đặc biệt quan trọng trong thống kê địa lý, nó thường được sử dụng để mô hình hóa độ cao trên bề mặt Trái đất với tập hợp các điểm đo.

Một trong những phép nội suy đa biến nổi tiếng là Bicubic, phép nội suy này là một phần mở rộng của phép nội suy bậc ba để nội suy các điểm dữ liệu trên lưới thông thường hai chiều. Hình ảnh bên là kết quả nội suy Bicubic trên hình vuông gồm 25 ô vuông đơn vị được vá lại với nhau. Màu sắc cho biết giá trị nội suy được. Các chấm đen là vị trí của dữ liệu có sẵn để nội suy.

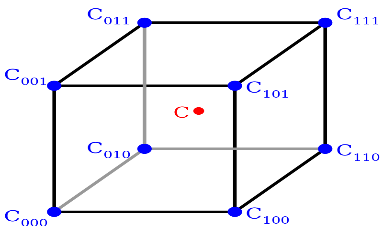
Hình 2: Ví dụ kết quả nội suy Bicubic

* 1. Lưới ba chiều (Regular three dimensional grid)

Nếu như lưới hai chiều là ma trận các hình vuông có tọa độ đỉnh (x,y) trong mặt phẳng thì lưới ba chiều là tập hợp các khối hộp có tọa độ đỉnh (x,y,z) trong không gian ba chiều.

Lưới lập phương ba chiều (3D Cartesian grid) là một dạng đặc biệt của lưới ba chiều, nó là tâp hợp các khối lập phương cơ sở. Mỗi khối lập phương cơ sở có cạnh bằng một đơn vị khoảng cách lưới. Với thiết kế kho lạnh thông thường có dạng hình hộp chữ nhật, chúng ta dễ dàng áp dụng dạng lưới này cho việc mô hình hóa nhiệt độ kho lạnh tại các điểm trong không gian thành các khối lập phương cơ sở. Mỗi khối lập phương sẽ được tô màu tương ứng với nhiệt độ mà nó biểu diễn.

Hình 3: Minh họa lưới lập phương ba chiều

* 1. Nội suy ba chiều (Trilinear interpolation)

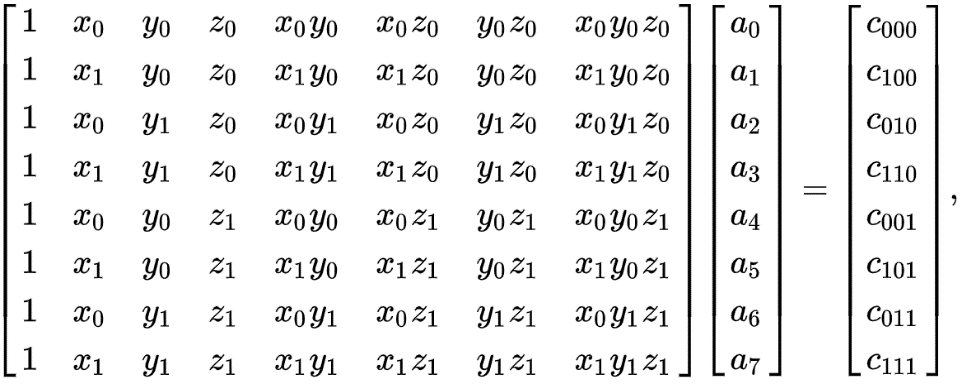
Nội suy ba chiều là một phương pháp nội suy đa biến trên lưới ba chiều. Phương pháp có đầu ra xấp xỉ giá trị của một hàm ƒ(x, y, z) tại một điểm cần xét trong khối hình hộp chữ nhật có dữ liệu tại tám điểm cần thiết. Tám điểm này là tám đỉnh của khối bao quanh điểm cần tính nội suy.

Hình 4: Minh họa khối nội suy

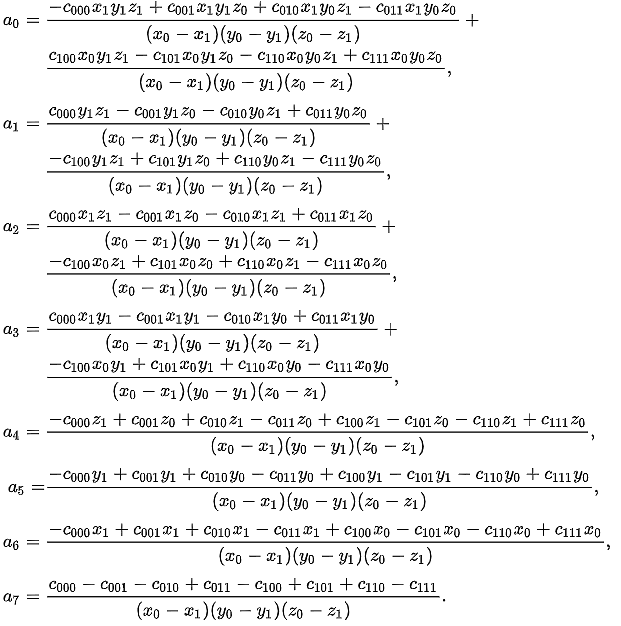
Dữ liệu của tám đỉnh này là: C000, C001, C010, C011, C100, C101, C110, C111.

Hàm nội suy cho kết quả:

trong đó các hệ số ai được tìm thấy bằng cách giải hệ thống tuyến tính.



*Với x0 là giá trị nhỏ nhất và x1 là giá trị lớn nhất của tọa độ trục x mà điểm cần xét có thể có trong khối lập phương (tương ứng xmin và xmax ). Điều này tương tự với y0,y1 và z0, z1 trên hai trục tọa độ còn lại.*

Mang lại kết quả:

1. Hàm nội suy ba chiều với dữ liệu cảm biến nhiệt độ trong kho lạnh
   1. Tập dữ liệu đầu vào

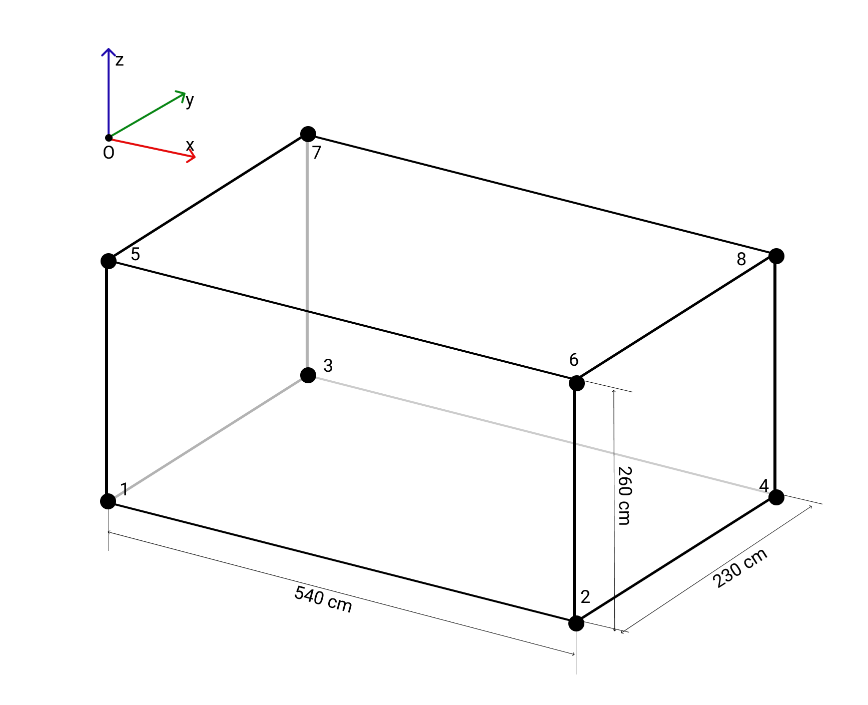
Tập dữ liệu đầu vào gồm:

* Cấu hình kho lạnh: chiều dài, chiều rộng, chiều cao và khoảng cách lưới lập phương ba chiều.
* Danh sách các cảm biến trong kho lạnh; mỗi cảm biến chứa vị trí tọa độ của nó theo lưới ba chiều trong kho lạnh và giá trị nhiệt độ đo được.

Dữ liệu đầu vào được xem là hợp lệ khi có đủ 8 điểm có vị trí tọa độ {C000, C001, C010, C011, C100, C101, C110, C111 } tương ứng với 8 gốc của kho lạnh, chúng tôi gọi các cảm biến ở các vị trí này là “cảm biến chính”.

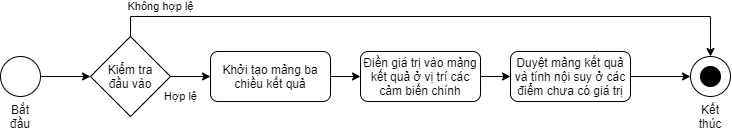
Tập dữ liệu đầu vào mẫu được mô tả như sau:

|  |  |  |  |
| --- | --- | --- | --- |
| **Bảng 1: Cấu hình kho lạnh (đơn vị cm)** | | | |
| Dài | Rộng | Cao | Khoảng cách lưới lập phương ba chiều |
| 540 | 230 | 260 | 10 |



Hình 5: Vị trí 8 cảm biến chính trong kho

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Bảng 2: Danh sách 8 cảm biến chính** | | | | | |
| STT | ID SENSOR | Tọa độ X | Tọa độ Y | Tọa độ Z | Giá trị (độ C) |
| 1 | 118 | 0 | 0 | 0 | -17.11 |
| 2 | 112 | 53 | 0 | 0 | -18.65 |
| 3 | 107 | 0 | 22 | 0 | -17.11 |
| 4 | 101 | 53 | 22 | 0 | -18.44 |
| 5 | 116 | 0 | 0 | 25 | -14.99 |
| 6 | 110 | 53 | 0 | 25 | -16.85 |
| 7 | 109 | 0 | 22 | 25 | -14.44 |
| 8 | 103 | 53 | 22 | 25 | -15.55 |

* 1. Lưu đồ thuật toán
  2. Kết quả đầu ra

Hàm nội suy ba chiều với dữ liệu cảm biến nhiệt độ kho lạnh cho kết quả đầu ra là mảng ba chiều A[x1][y1][z1]. Với dữ liệu đầu vào mẫu bên trên thì chúng ta có mảng kết quả ba chiều A[53][23][24] tương ứng 29.256 giá trị nhiệt độ trong kho lạnh ở các điểm tọa độ khác nhau.

Để kiểm tra sự chính xác của hàm nội suy, chúng tôi đặt vào kho lạnh các cảm biến ở các vị trí sau: P1(26,0,0), P2(26,22,0), P3(26,0,25), P4(26,22,25). Ghi nhận kết quả đo của cảm biến và so sánh giá trị nội suy được. Chúng tôi nhận thấy độ chênh lệch giữa 2 giá trị này không lớn, nên hàm nội suy được sử dụng bên trên có thể áp dụng vào mô hình hóa nhiệt độ kho lạnh.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Bảng 3: Kiểm tra kết quả hàm nội suy và giá trị đo thực tế** | | | | | | | | |
|  | **P1 (ID: 115)** | | **P2 (ID: 104)** | | **P3 (ID: 113)** | | **P4 (ID: 106)** | |
| **Lần đo** | **Giá trị đo** | **Giá trị nội suy** | **Giá trị đo** | **Giá trị nội suy** | **Giá trị đo** | **Giá trị nội suy** | **Giá trị đo** | **Giá trị nội suy** |
| 1 | -17.32 | -17.34 | -16.36 | -16.41 | -15.88 | -16.06 | -15.32 | -15.41 |
| 2 | -17.44 | -17.47 | -16.94 | -16.97 | -15.88 | -15.90 | -15.44 | -15.64 |
| 3 | -17.65 | -17.83 | -16.79 | -16.97 | -14.32 | -14.52 | -15.88 | -15.88 |
| 4 | -17.98 | -18.16 | -15.79 | -15.94 | -12.44 | -12.56 | -13.66 | -13.75 |
| 5 | -16.00 | -16.02 | -15.33 | -15.34 | -12.00 | -12.12 | -13.10 | -13.27 |
| 6 | -16.00 | -16.08 | -15.58 | -15.69 | -13.88 | -14.06 | -14.66 | -14.80 |
| Độ sai lệch | 0.085 | | 0.09 | | 0.14 | | 0.12 | |

1. Phương pháp thu hẹp phạm vi cho hàm nội suy dựa vào vị trí các cảm biến phụ trong kho lạnh.

Một kho lạnh cần tối thiểu 8 cảm biển chính để có thể nội suy được nhiệt độ ở các khu vực trong kho lạnh. Tuy nhiên, với nhu cầu tăng độ chính xác cho hàm nội suy và có thể giám xác nhiệt độ ở các điểm quan trọng trong kho lạnh thì người dùng có thể lắp thêm nhiều cảm biến phụ tại bất kì một vị trí nào khác. Để giải quyết vấn đề này, chúng tôi đề xuất một giải pháp phân chia khu vực nội suy từ toàn bộ kho lạnh thành nội suy các khối con. Các khối con này được tách ra từ khối chính theo vị trí của các cảm biến phụ với chiến lược ưu tiên tách các cảm biến gần tâm khối chính trước. Số lượng khối con thu được cũng phụ thuộc vào vị trí cảm biến phụ hiện hữu.

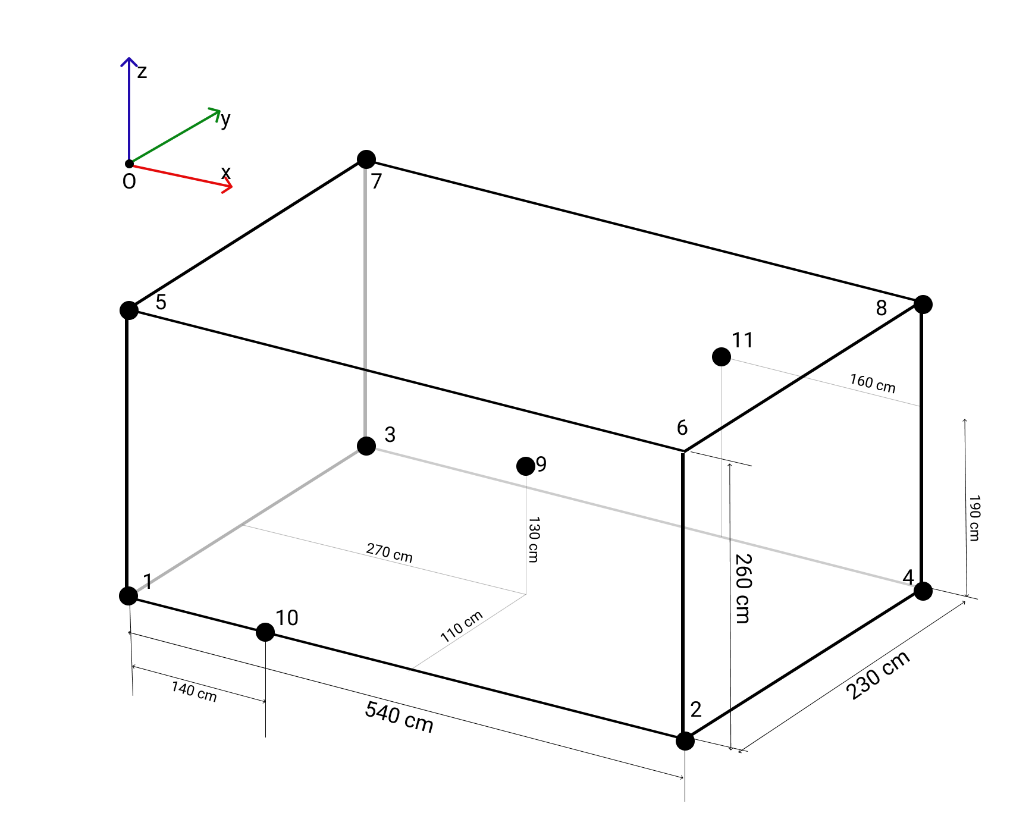
* 1. Tập dữ liệu đầu vào

Thêm vào tập dữ liệu mẫu ban đầu 3 cảm biến (9,10,11) ta được tập dữ liệu mẫu mới như sau:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Bảng 4: Danh sách cảm biến sau khi thêm 3 cảm biến phụ** | | | | | |
| **STT** | **ID SENSOR** | **Tọa độ X** | **Tọa độ Y** | **Tọa độ Z** | **Giá trị (độ C)** |
| 1 | 118 | 0 | 0 | 0 | -17.11 |
| 2 | 112 | 53 | 0 | 0 | -18.65 |
| 3 | 107 | 0 | 22 | 0 | -17.11 |
| 4 | 101 | 53 | 22 | 0 | -18.44 |
| 5 | 116 | 0 | 0 | 25 | -14.99 |
| 6 | 110 | 53 | 0 | 25 | -16.85 |
| 7 | 109 | 0 | 22 | 25 | -14.44 |
| 8 | 103 | 53 | 22 | 25 | -15.55 |
| **9** | 119 | 26 | 11 | 12 | -16.36 |
| **10** | 120 | 13 | 0 | 0 | -16.54 |
| **11** | 121 | 37 | 22 | 18 | -16.32 |

Chúng tôi thử nghiệm đặt các cảm biến phụ này theo các vị trí đặc biệt như sau:

* Cảm biến số **9** nằm bên trong khối chính, có vị trí gần tâm khối chính nhất.
* Cảm biến số **10** nằm trên một cạch của khối chính, ở đây là cạnh x
* Cảm biến số **11** nằm trong một mặt của khối chính.



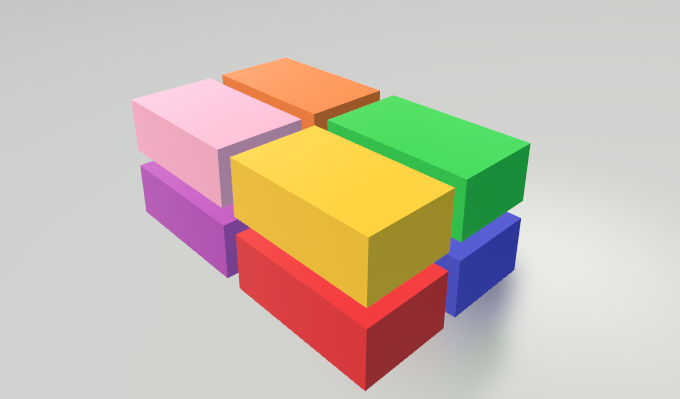
Hình 6: Vị trí cảm biến trong kho khi thêm 3 cảm biến phụ

* 1. Thuật toán đề xuất

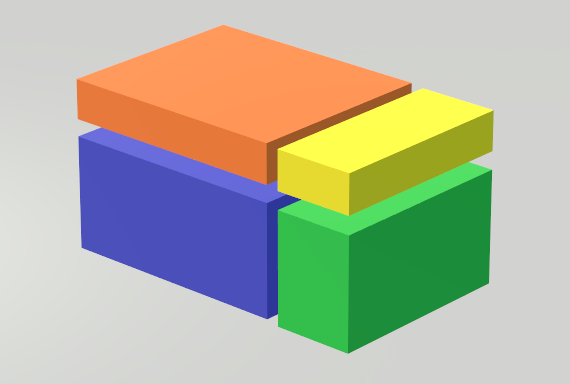
Dựa vào 3 trường hợp đặc biệt của vị trí cảm biến phụ, chúng tôi thực hiện tách khối chính ban đầu với 3 cách theo thứ tự ưu tiên như sau:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Bảng 5: Trường hợp tách khối** | | | | | |
| **Độ ưu tiên** | **Trường hợp** | **Tọa độ cảm biến phụ** | | | **Số khối tách ra được từ khối chính** |
| **x** | **y** | **z** |
| 1 | Cảm biến phụ nằm bên trong khối chính | x0 < x < x1 | y0 < y < y1 | z0 < z < z1 | 8 |
| 2 | Cảm biến phụ nằm trên một mặt bất kì của khối chính | x0 = x || x = x1 | y0 < y < y1 | z0 < z < z1 | 4 |
| x0 < x < x1 | y0 = y || y = y1 | z0 < z < z1 |
| x0 < x < x1 | y0 < y< y1 | z0 = z || z = z1 |
| 3 | Cảm biến phụ nằm trên một cạnh bất kì của khối chính | x0 < x < x1 | y0 = y || y = y1 | z0 = z || z = z1 | 2 |
| x0 = x || x = x1 | y0 < y < y1 | x0 = x || x = x1 |
| x0 = x || x = x1 | y0 = y || y = y1 | z0 < z < z1 |

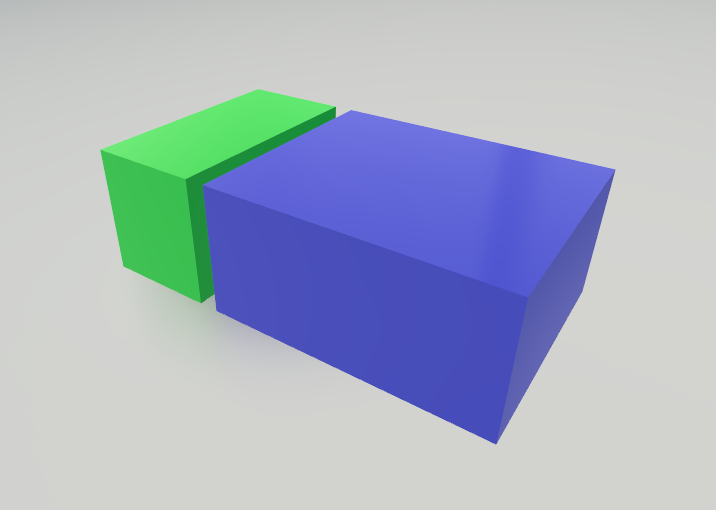
Ví dụ minh họa cách tách khối chính thành các khối con:

* Điểm tách ở vị trí cảm biến số 9:

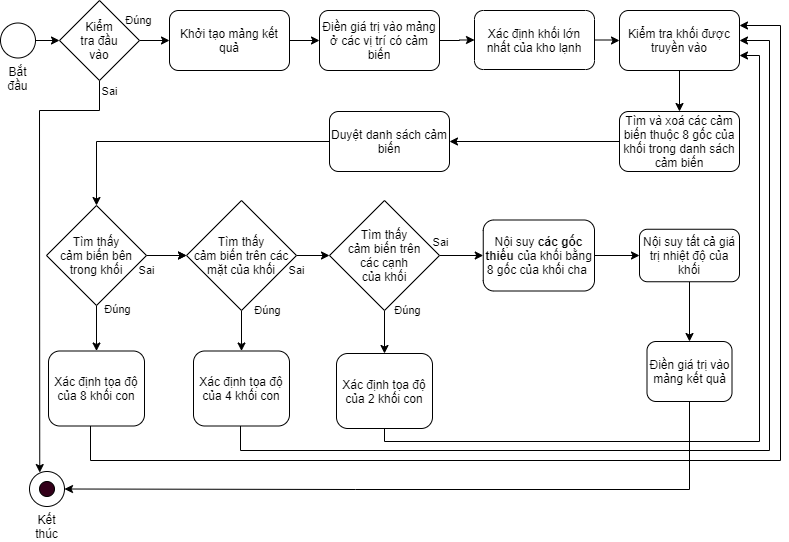
Hình 7: Minh tọa tách khối chính thành 8 khối con

* Điểm tách ở cảm biến số 11:

Hình 8: Minh tọa tách khối chính thành 4 khối con

* Điểm tách ở cảm biến số 10:

Hình 9: Minh tọa tách khối chính thành 2 khối con

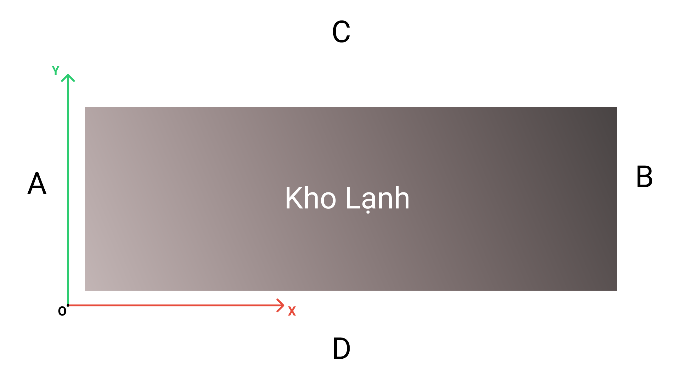
Lưu đồ thuật được cải tiến so với ban đầu như sau:

* 1. Kết quả

Thực nghiệm với dữ liệu mẫu bên trên cho thấy, việc sử dụng phương pháp chia tách kho lạnh giúp phát hiện tốt hơn các điểm nhiệt độ bất thường ở các vị trí quan trọng cần giám sát. Các cảm biến đặt ở vị trí bên trong của khối chính cho thấy khả năng phát hiện điểm nhiệt lạ tốt hơn các cảm biến ở vị trí khác.

1. THƯ VIỆN ĐỒ HỌA 3D THREE JS
2. Mô tả vấn đề
   1. Dữ liệu đầu vào
      1. Dữ liệu cấu hình kho lạnh

Dữ liệu cấu hình kho lạnh là một đối tượng có ba thuộc tính:

* **size**: là đối tượng quy định kích thước của kho lạnh, gồm bốn thuộc tính:
* **x**: số khối tối đa theo chiều rộng của kho lạnh.
* **y**: số khối tối đa theo chiều dài của kho lạnh.
* **z**: số khối tối đa theo chiều cao của kho lạnh.
* **tilesize**: kích thước của một đơn vị thể hiện trên mô hình, đơn vị là điểm ảnh (px).
* **door**: là đối tượng cấu hình cửa, gồm hai thuộc tính:
* **show**: quyết định có hiển thị cửa trên mô hình hay không. Nếu *“show = true”* thì hiển thị cửa trên mô hình, ngược lại *“show = false”* thì không hiển thị.
* **direction**: mô tả vị trí cửa của kho lạnh theo có 4 hướng: A, B, C, D. Các hướng này được định vị theo trục tọa độ như sau:

Hình 10: Mô tả vị trí cửa

* **axis-labels**: mô tả các nhãn được gắn trên ba trục tọa độ.
* **axis-x**: mô tả các nhãn trên trục x.
  + - **show**: xác định xem có hiển thị các nhãn trên trục x hay không.
    - **list**: mảng các số nguyên chứa tọa độ của các nhãn trên trục x.
* **axis-y**: mô tả các nhãn trên trục y.
  + - **show**: xác định xem có hiển thị các nhãn trên trục y hay không.
    - **list**: mảng các số nguyên chứa tọa độ của các nhãn trên trục y.
* **axis-z**: mô tả các nhãn trên trục z.
  + - **show**: xác định xem có hiển thị các nhãn trên trục z hay không.
    - **list**: mảng các số nguyên chứa tọa độ của các nhãn trên trục z.
    1. Dữ liệu nhiệt độ

Dữ liệu nhiệt độ là một đối tượng có ba thuộc tính như sau:

* **values**: là một mảng ba chiều chứa thông tin nhiệt độ của kho lạnh, mảng này được tính toán dựa vào phương pháp nội suy mà chúng tôi sử dụng ở trên. Mảng có dạng A[x1][y1][z1]. Nhiệt độ ở một điểm P(x,y,z) sẽ tương ứng phần tử A[x][y][z] của mảng. Ví dụ: nhiệt độ ở vị trí P(1,2,5) = A[1][2][5].
* **min**: nhiệt độ thấp nhất trong kho lạnh.
* **max**: nhiệt độ cao nhất trong kho lạnh.
  + 1. Dữ liệu cấu hình lát cắt

Dữ liệu cấu hình lát cắt là một đối tượng có hai thuộc tính:

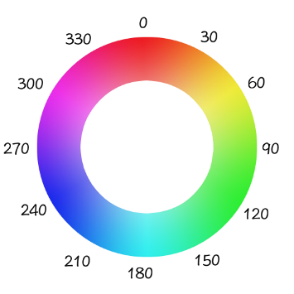
* **axis**: chứa tên trục muốn cắt, là một ký tự viết thường (x, y hoặc z).
* **level**: chứa một số nguyên, là số lớp muốn cắt.
  1. Kết quả đầu ra mong muốn
* Hiển thị khối hình hộp với các ô màu tương ứng với dữ liệu nhiệt độ đã cung cấp trước đó.
* Có thể xoay 360 độ, không có ô nào bị che khuất.
* Có thể cắt lớp khối kho lạnh theo ba trục x, y và z bằng giao diện đồ họa.
* Có thể hiển thị các nhãn đã cài đặt trên ba trục tọa độ, vị trí chính xác, nội dung rõ ràng.

1. Giới thiệu thư viện three.js

Three.js là một thư viện JavaScript 3D dễ sử dụng, gọn nhẹ. Nó cung cấp nhiều trình kết xuất như canvas, WebGL, v.v.

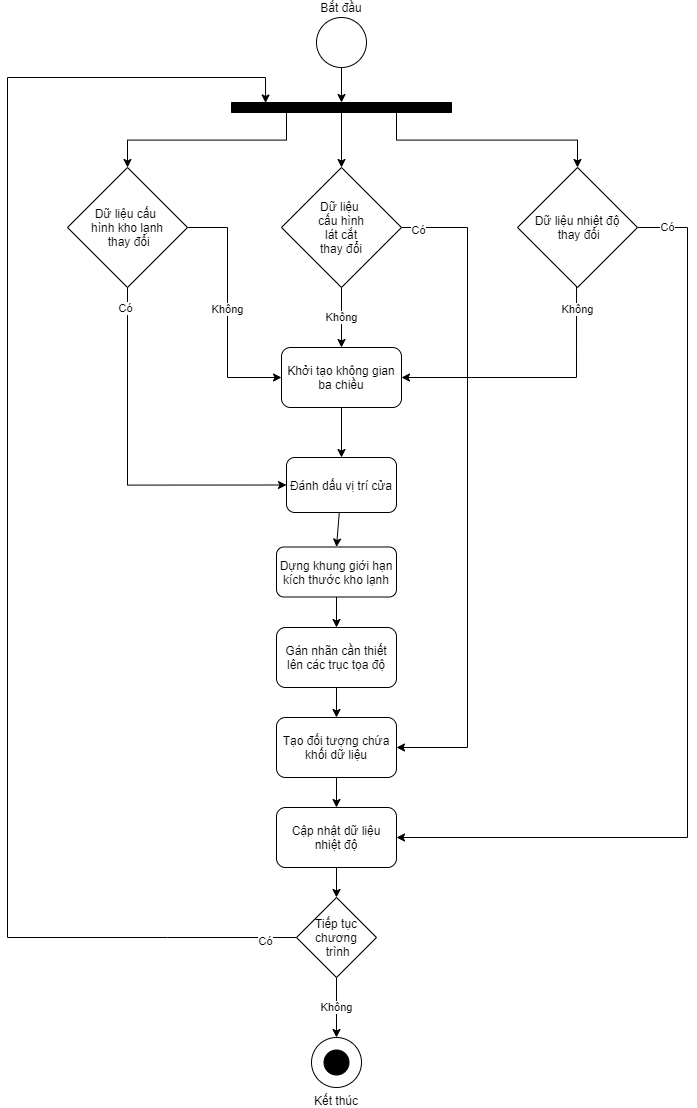
WebGL (Thư viện đồ họa Web) là một API JavaScript mang đồ họa 3D được tăng tốc phần cứng cho trình duyệt mà không cần sự trợ giúp của các tiện ích bổ sung nào khác. WebGL chạy trên GPU (Bộ xử lý đồ họa) của card đồ họa của bạn, điều đó có nghĩa là CPU của bạn (Bộ xử lý trung tâm) có thể tập trung vào các tác vụ khác. Từ quan điểm của người dùng, các ứng dụng chỉ đơn giản là chạy nhanh hơn nhiều. Thư viện Three.js giúp làm việc với WebGL dễ dàng hơn.

1. Hệ màu HSL

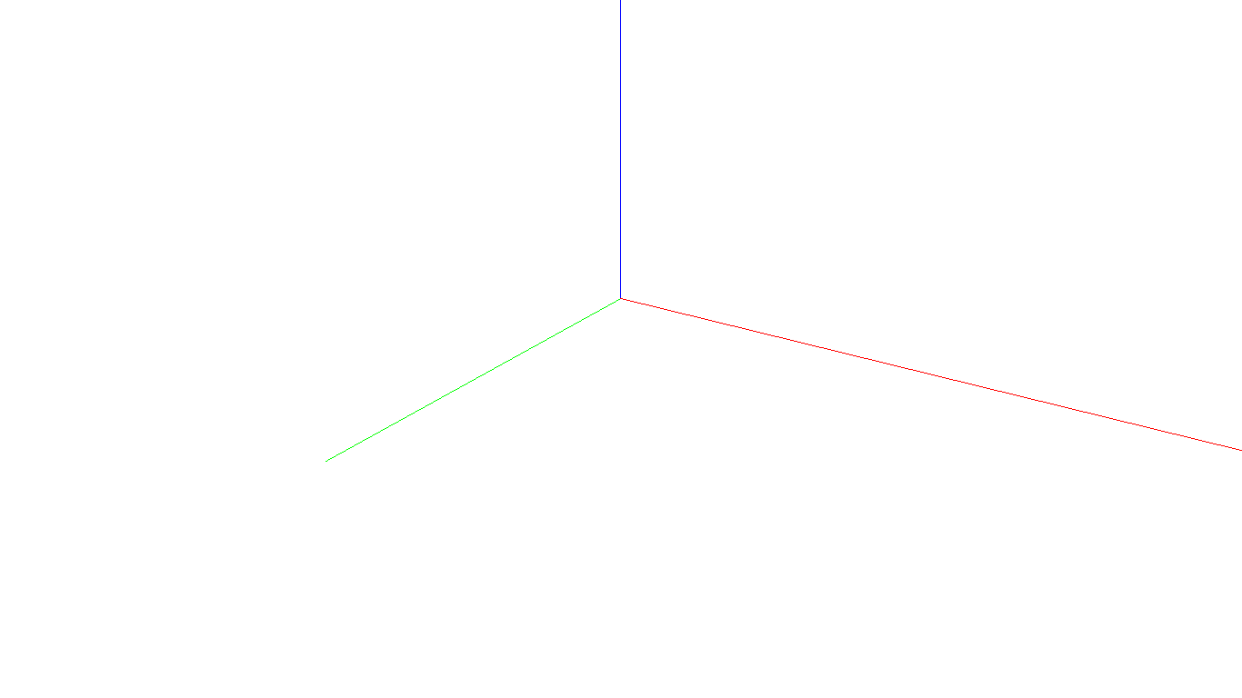
Không gian màu HSL, còn gọi là không gian màu HSV hay HSB, là một không gian màu dựa trên ba số liệu:

Hình 11: Minh họa mã màu HSL

* **H**: (Hue) vùng màu hay còn gọi là góc độ màu.
* **S**: (Saturation) độ bão hòa màu,
* **L** (hay V, B): (Lightness, Value hay Bright) độ sáng.

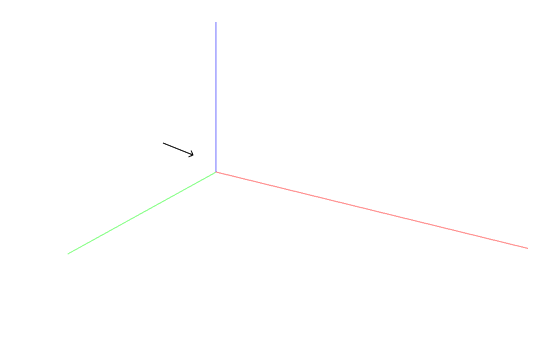
1. Các bước dựng mô hình phân bố nhiệt độ trong không gian 3 chiều
   1. Khởi tạo không gian 3 chiều

Khởi tạo các đối tượng cơ bản cần có của không gian ba chiều trong thư viện Three.js như:

* **Scene**: màn ảnh, chứa tất cả các đối tượng 3D bên trong.
* **Camera**: máy quay, chiếu góc hình lên màn ảnh, cho ra hình ảnh tại một góc nhìn xác định.
* **Renderer**: đối tượng vẽ hình chiếu ra bên ngoài.
* **OrbitControls**: cung cấp các tính năng điều khiển, xoay góc nhìn xung quanh mô hình.
* **AxesHelper**: vẽ các trục tọa độ lên màn ảnh.

Hình 12:Khởi tạo trục tọa độ

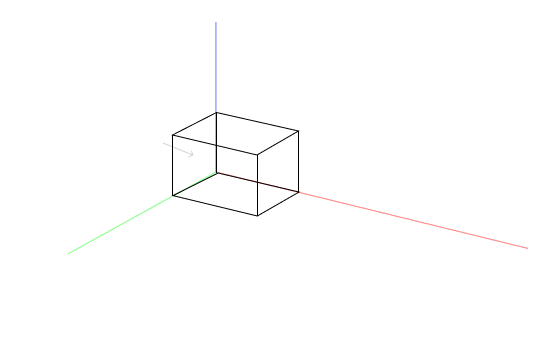
* 1. Đánh dấu vị trí cửa

Đọc cấu hình kho lạnh được truyền vào, tính toán độ lớn và vị trí cửa. Tạo đối tượng **ArrowHelper** với các thông số đã tính toán để thể hiện vị trí cửa.

Hình 13: Đánh dấu vị trí cửa

* 1. Dựng khung giới hạn kích thước kho lạnh

Đọc cấu hình được truyền vào, tính toán kích thước, vị trí các nút và dựng các đoạn thẳng tại các vị trí thích hợp dựa trên dữ liệu đã đọc.



Hình 14: Dựng khung giới hạn kho lạnh

* 1. Tính kích thước của khối dữ liệu cần hiển thị

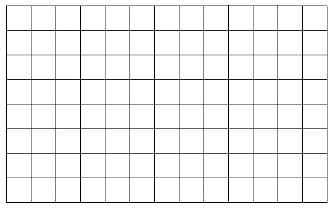
Đọc cấu hình được truyền vào, bao gồm cấu hình lát cắt từ đó tính ra kích thước mới .

* 1. Tạo đối tượng chứa khối dữ liệu

Thực hiện các công việc sau:

* Căn chỉnh trung tâm của máy quay tại trọng tâm của khối hộp.
* Tạo các khung mặt phẳng của khối hộp bằng phương thức **createAFace**.
* Gộp sáu đối tượng đã tạo vào một đối tượng **Object3D**.
* Thêm đối tượng **Object3D** vừa rồi vào màn ảnh.

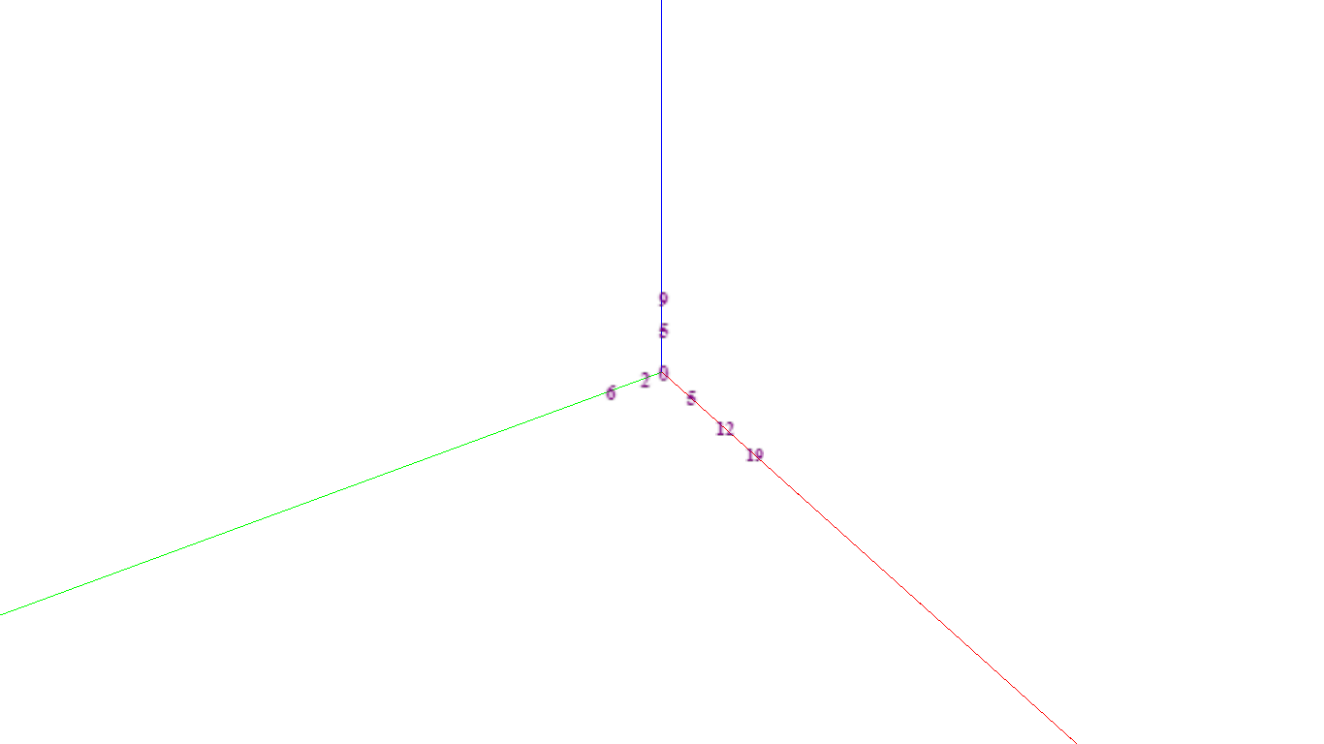
**Phương thức createAFace gồm các bước như sau:**

* + Xác định kích thước khối hộp sắp tạo.
  + Xác định thứ tự của mặt cần tạo.

Hình 15: Minh họa một mặt của khối kho lạnh

* + Xác định tọa độ, kích thước và góc quay dựa trên kích thước và thứ tự.
  + Tạo hình dạng của mặt phẳng.
  + Tạo vật liệu của mặt phẳng.
  + Tạo đối tượng vật thể của mặt phẳng.
  + Loại bỏ các đối tượng không cần thiết
  + Trả về vật thể vừa tạo.
  1. Dán các nhãn cần thiết lên các trục tọa độ

Gồm các bước:

* Đọc dữ liệu cấu hình
* Duyệt qua các tọa độ cần gắn nhẵn
  + Tạo đối tượng Sprite (luôn đối mặt với camera).
  + Tính toán tọa độ.
  + Thêm đối tượng Sprite vừa tạo vào tọa độ thích hợp.

Hình 16: Đánh số trên trục tọa độ

* 1. Cập nhật dữ liệu nhiệt độ

Duyệt qua mảng dữ liệu, kiểm tra xem phần tử hiện tại có thuộc vào mặt phẳng nào của hình hộp không. Nếu có thì tính sắc độ và cập nhật màu của mô hình tại tọa độ đó, ngược lại thì bỏ qua phần tử đó.

* + 1. Cập nhật dữ liệu nhiệt độ tại một tọa độ

Hàm nhận vào hai giá trị lần lượt là thứ tự mặt phẳng và nhiệt độ tại điểm cần cập nhật theo thứ tự từ trên xuống dưới, từ trái sang phải của mặt phẳng.

* + 1. Chuyển nhiệt độ sang sắc độ

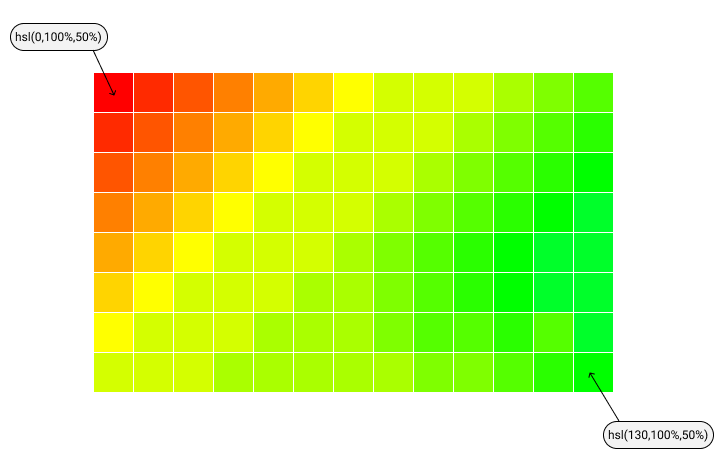
Hàm nhận vào ba giá trị lần lượt là:

* **min**: nhiệt độ thấp nhất.
* **max**: nhiệt độ cao nhất.
* **temp**: nhiệt độ của ô cần tính.

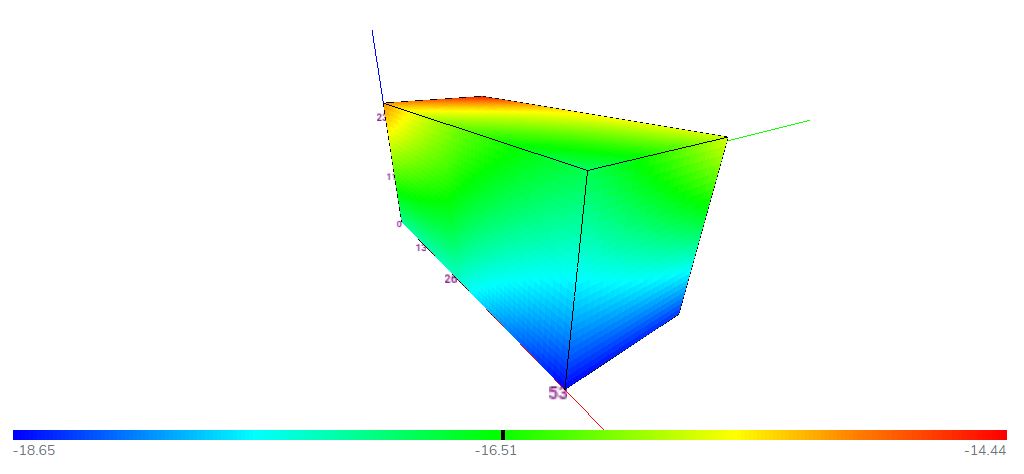
Giá trị **H** của hệ màu **HSL** trả về được tính theo công thức:

Giá trị mặc định của **S** và **L** lần lược là **100%** và **50%**, chúng tôi không thay đổi giá trị này. Do đó mã màu **HSL** tại một điểm nhiệt độ là: **hsl(H,100%,50%)**.

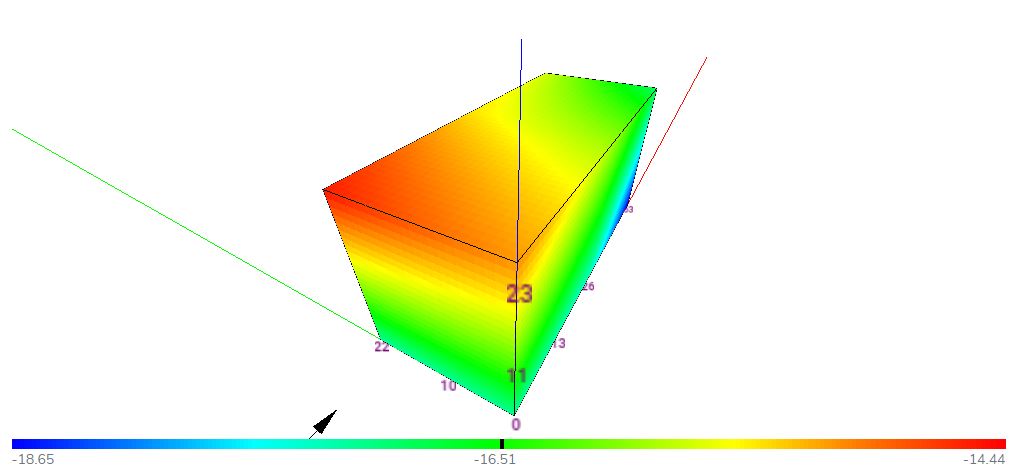
Hình 17: Minh họa màu của một mặt khối kho lạnh sau khi cập nhật giá trị nhiệt độ

1. Kết quả thực Bảng 2nghiệm với dữ liệu thực tế

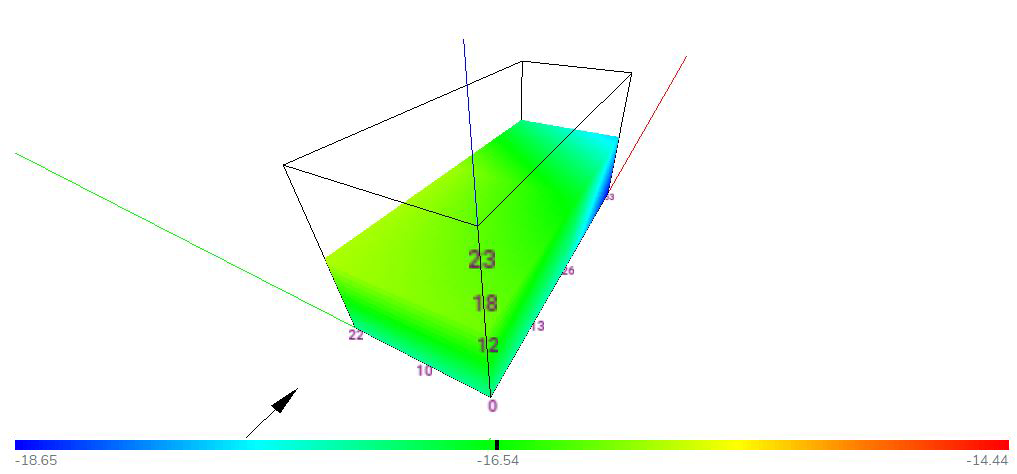
Dựng mô hình 3D với dữ liệu mẫu *(Bảng 4: Danh sách cảm biến sau khi thêm 3 cảm biến phụ).* Chúng tôi thu được mô hình khối nhiệt độ kho lạnh như sau:



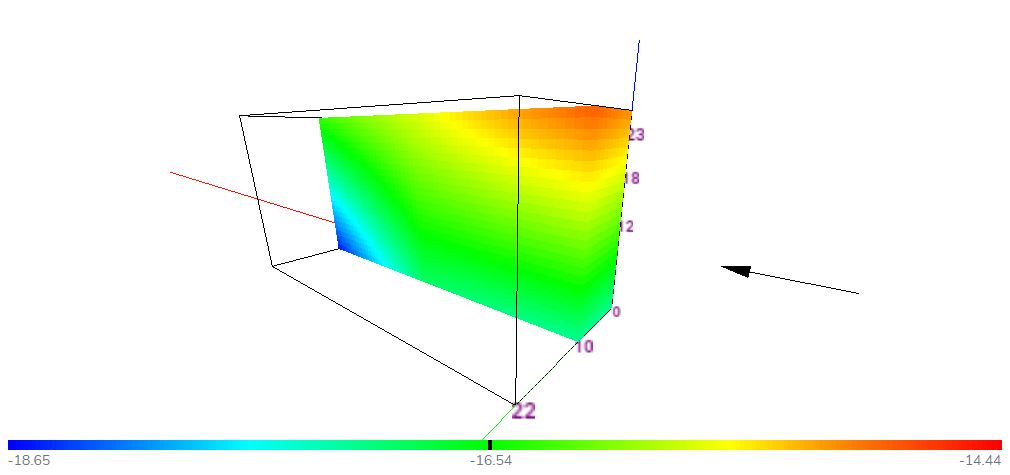
Hình 18: Mô hình khối nhiệt độ kho lạnh ở điểm nhiệt độ thấp nhất



Hình 19: Mô hình khối nhiệt độ kho lạnh ở điểm nhiệt độ cao nhất



Hình 20: Mô hình khối nhiệt độ kho lạnh cắt lớp z:18



Hình 21: Mô hình khối nhiệt độ kho lạnh cắt lớp y:10