

# 动态色度标定的边界处理

## 一、意义

动态色度标定在方位测井识别时常用的处理方法，但是传统的动态色度标定的处理方案存在连续性不好的问题，在边界上经常会生成一个明显的边界，这样的边界容易影响解释人员的对于地层划分的判断。

## 二、数据预处理

### 1.平滑滤波

#### API

```
def gaussianFilter(data:pd.DataFrame,sigma=3)
```

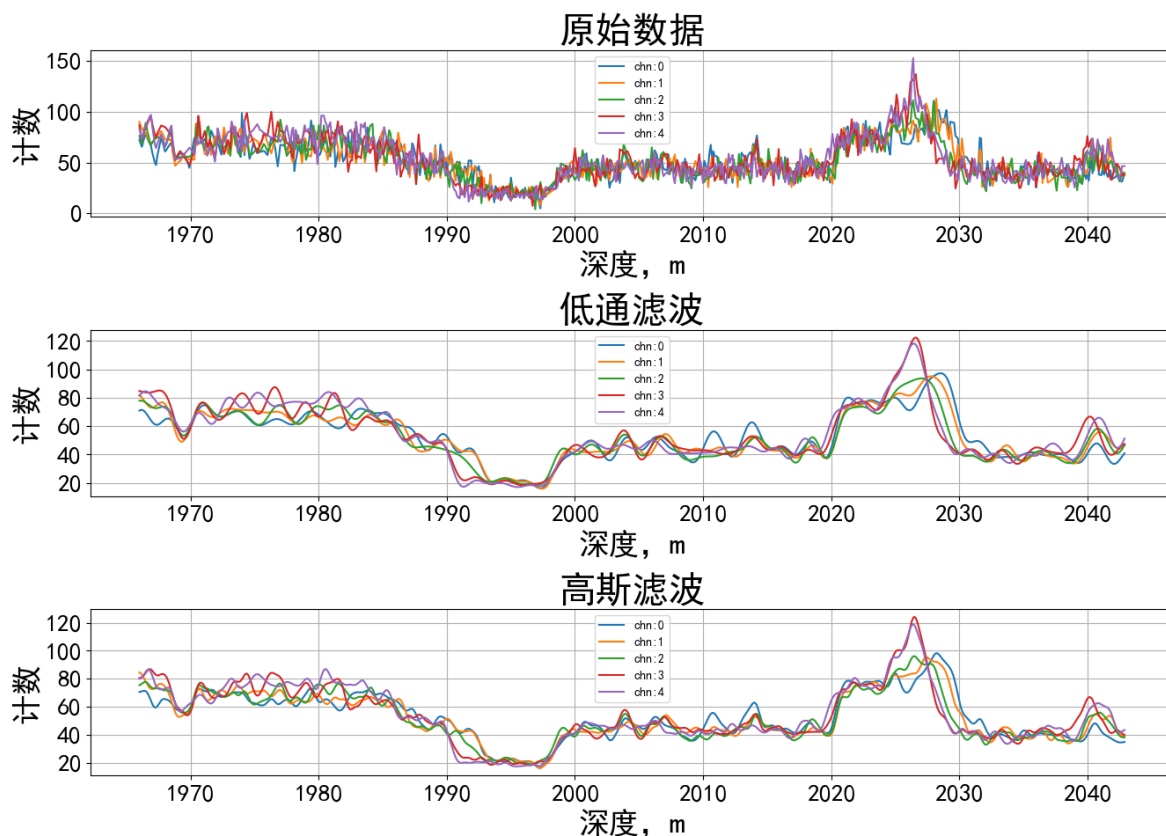
- 输入参数

变量名	type	说明
data	pandas.DataFrame	index为深度值，标签为“DEPTH”；每一列为不同方位的数据，列标签不做要求
sigma	int	高斯核函数的标准差

- 返回值
  - **type:** pandas.DataFrame
  - index为深度值，标签为“DEPTH”；每一列为不同方位的数据，列标签与输入值相同

#### 函数说明

通过滤波操作对数据进行平滑操作，减少噪声对解释人员的干扰。常用滤波方法有低通滤波、均值滤波、高斯滤波等，这里展示低通滤波和高斯滤波



从平滑效果来说，低通滤波的平滑效果更好，高斯滤波的结果仍保留一定的低频波动。但是由于低通滤波存在卷积操作，实际工程中，由于无法实时填补缺失值，此时无法利用低通滤波来处理，因此数据利用高斯滤波处理来模拟实际工程中的应用场景

## 2.插值

### API

```
def interpolation(data:pd.DataFrame,kind="cubic",tarNum=720)
```

- 输入参数

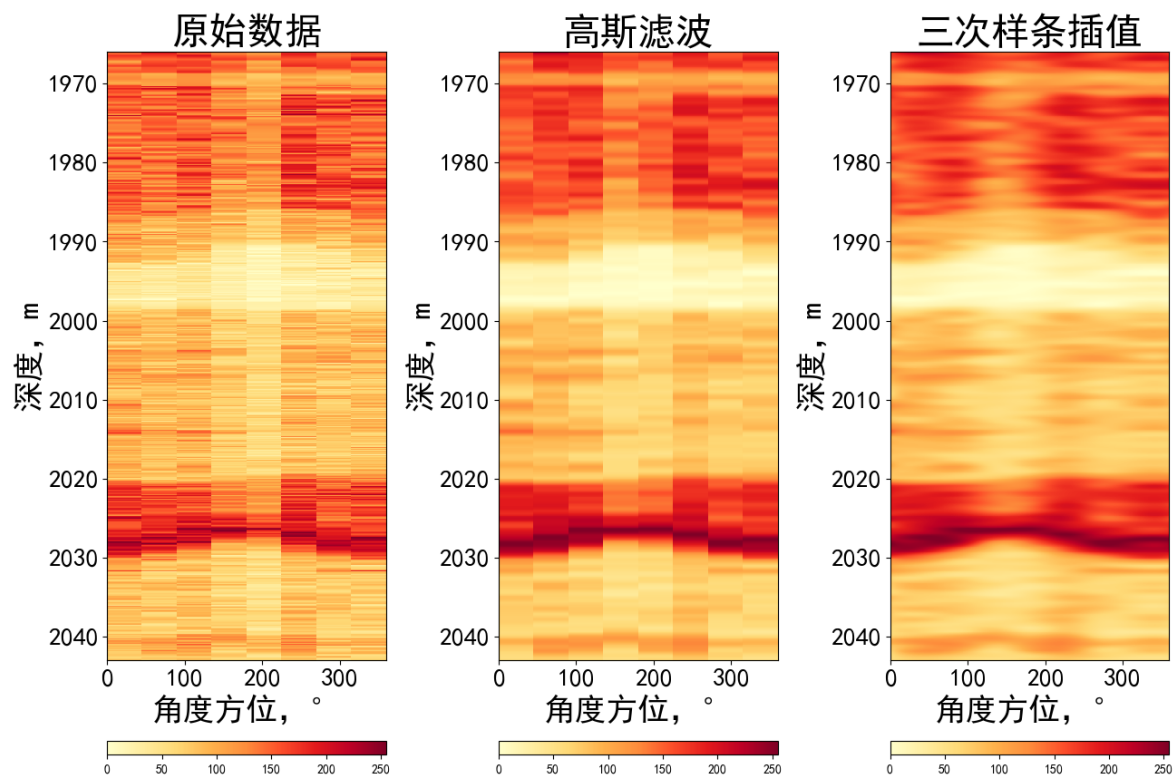
变量名	type	说明
data	pandas.DataFrame	index为深度值，标签为“DEPTH”；每一列为不同方位的数据，列标签不做要求
kind	“linear”、“quadratic”、“cubic”中的一个	插值方法：“linear”：线性插值；“quadratic”：二阶样条插值；“cubic”：三次样条插值
tarNumber	int	插值后的列数

- 返回值

- **type:** pandas.DataFrame
- index为深度值，标签为“DEPTH”；每一列为不同方位的数据，列标签变为角度值（单位：°，从0°到360°）

函数说明

由于方位测井数据在角度方向上的数据点较少，导致角度方向的分辨率较低，因此我们采用三次样条插值的方法，提高角度方向上的数据点数量。插值方法选用三次样条插值方法，将8列角度方位数据拓展到360列。



三、经典色度标定方法

1.静态色度标定

直方图均衡

```
def histogramEqualization(data:pd.DataFrame)
```

- 输入参数

变量名	type	说明
data	pandas.DataFrame	index为深度值，标签为“DEPTH”；每一列为不同方位的数据，列标签不做要求

- 返回值
  - type: pandas.DataFrame
  - index为深度值，标签为“DEPTH”；每一列为不同方位的数据，列标签与输入值相同

变换函数：直方图均衡

```
def transform_HE(data:pd.DataFrame)
```

- 输入参数

变量名	type	说明
data	pandas.DataFrame	index为深度值，标签为“DEPTH”；每一列为不同方位的数据，列标签不做要求

- 返回值
  - type: function
  - 返回一个直方图变换函数function(x)，等同于参数确定的  
`histogramEqualization(data:pd.DataFrame)`

## 函数说明

在测井成像时，需要将测井数据转换成色标值在计算机上显示，由此色度标定测井成像中非常关键的环节。静态色度标定是根据在整个井段的数据分布得到一个映射函数，函数可以表示如下：

$$f(V_i) = V_i * S + V_{offset} \quad (1)$$

其中S与V\_{offset}是刻度系数和偏移量，这些参数可以根据井段数据的最大值V\_{max}和最小值V\_{min}、色标最小值C\_{min}和色标最大值C\_{max}求得：

$$S = (C_{max} - C_{min}) / (V_{max} - V_{min}) \quad (2)$$

$$V_{offset} = C_{min} - V_{min} * S \quad (3)$$

实际使用的函数只需要将数据映射到0-255的灰度表上，从灰度表映射到色标由绘图软件自行完成。

## 2.动态色度标定

### API

```
def dynamicOperation(data:pd.DataFrame,function: callable,winlen=300)
```

- 输入参数

变量名	type	说明
data	pandas.DataFrame	index为深度值，标签为“DEPTH”；每一列为不同方位的数据，列标签不做要
function	callable(函数)	处理每一个数据窗数据的函数（如 <code>histogramEqualization(data:pd.DataFrame)</code> ）
winlen	int	窗长

- 返回值
  - **type:** pandas.DataFrame
  - index为深度值，标签为“DEPTH”；每一列为不同方位的数据，列标签与输入值相同

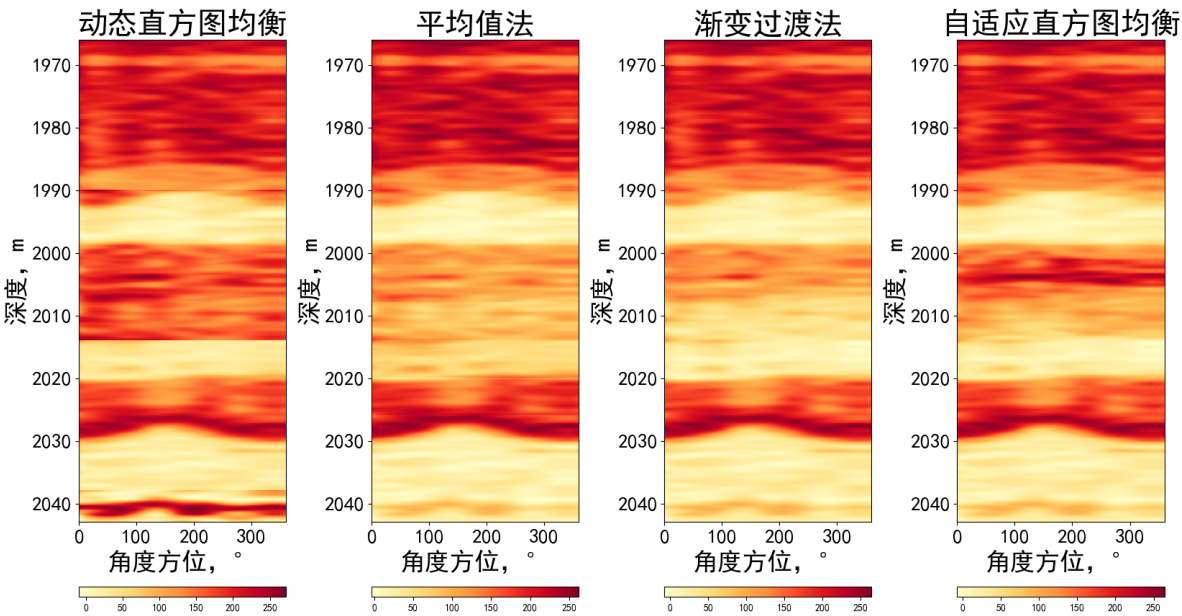
函数说明

静态色度标定相当于基于全部井段数据进行一次直方图均衡化处理，然而这种处理方式在数据分布不均匀时，即使进行了直方图均衡化仍然会有大量数据堆积在一小段区域，从而使图像的局部细节特征不明显。针对这个问题，赖富强提出了动态色度标定法。

动态色度标定法首先需要选取一个滑动窗长，在窗长范围内进行静态色度标定，然后在整个处理井段上按照一定步长移动窗口（步长一般取窗长的4/5），重复做静态色度标定直到完成整个井段的色度标定。

四、动态色度标定的改进方案

由于动态色度标定在窗口的边界连续性不好，容易生成色标值的阶跃断层，我们需要对动态色度标定进行一些改进来消除其阶跃断层。西安科技大学的唐佳伟提出了自适应窗长法来减弱动态色度产生的断层。这里我们提出了几种与自适应窗长法不同的方法来减弱其产生的断层现象：



1.平均值法

API

```
def dynamicOperation_MeanValue(data:pd.DataFrame,function: callable,winlen=300)
```

- 输入参数

变量名	type	说明
data	pandas.DataFrame	index为深度值，标签为“DEPTH”；每一列为不同方位的数据，列标签不做要
function	callable(函数)	处理每一个数据窗数据的函数（如 <code>histogramEqualization(data:pd.DataFrame)</code> ）
winlen	int	窗长

- 返回值
  - **type:** pandas.DataFrame

- index为深度值，标签为“DEPTH”；每一列为不同方位的数据，列标签与输入值相同

### 函数说明

平均值法与经典的动态色度标定法类似，首先选取一个滑动窗长，不断移动窗口并在窗口中进行静态色度标定。与之不同之处在于，平均值法一般选取窗长1/5为滑动步长，滑动处理完整个井段数据后，每段数据都经历了五次静态色度标定（开始部分和结束部分少于5次），最终结果由这五次静态色度标定的加权平均值决定。

测井数据	第1次标定	第2次标定	第3次标定	第4次标定	第5次标定	第6次标定	第7次标定
1	A1						
2	A2	B1					
3	A3	B2	C1				
4	A4	B3	C2	D1			
5	A5	B4	C3	D2	E1		
6		B5	C4	D3	E2	F1	
7			C5	D4	E3	F2	G1
8				D5	E4	F3	G2
9					E5	F4	G3
10						F5	G4
11							G5

对于第五段的数据由以下公式计算：

$$V_{result} = \frac{\begin{bmatrix} w_1 & w_2 & w_3 & w_4 & w_5 \end{bmatrix} \cdot \begin{bmatrix} A_5 & B_4 & C_3 & D_2 & E_1 \end{bmatrix}}{\begin{bmatrix} w_1 & w_2 & w_3 & w_4 & w_5 \end{bmatrix} \cdot \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \end{bmatrix}}$$

w\_i为权值，一般采用单位向量或者高斯滤波算子来对于各次的色度标定加权，通过平均值法可以有效的减弱阶跃断层的程度。如使用高斯滤波算子距这段数据越近的数据段的数据分布范围对其影响越大，相较于单位向量可以更好的显现测井数据的局部特征。

## 2.渐变过渡法

### API

```
def dynamicOperation_GradualChange(data:pd.DataFrame,function, winlen:int = 300)
```

- 输入参数

变量名	type	说明
data	pandas.DataFrame	index为深度值，标签为“DEPTH”；每一列为不同方位的数据，列标签不做要
function	callable(函数)	处理每一个数据窗数据的函数（如 <code>histogramEqualization(data:pd.DataFrame)</code> ）
winlen	int	窗长

- 返回值
  - **type:** pandas.DataFrame
  - index为深度值，标签为“DEPTH”；每一列为不同方位的数据，列标签与输入值相同

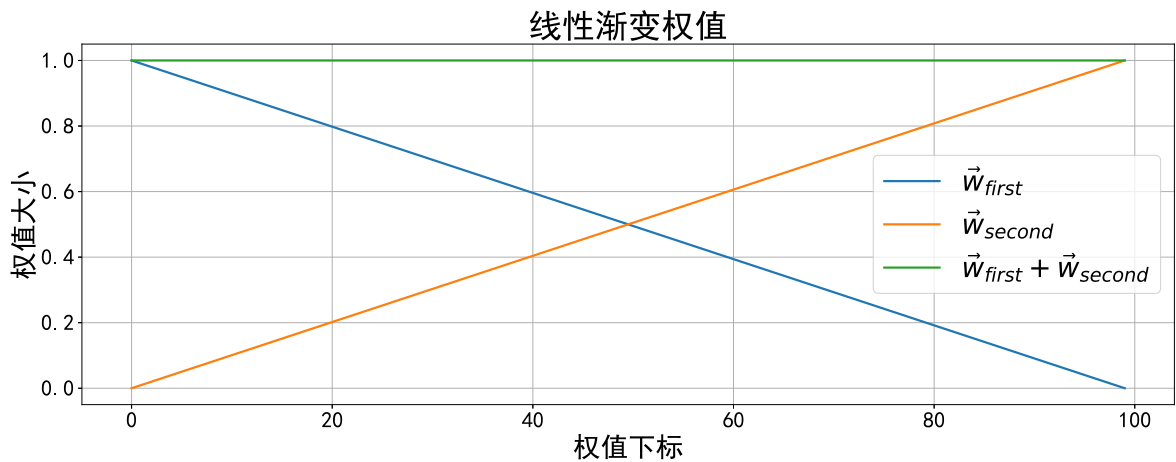
### 函数说明

渐变过渡法并不改变色度标定的窗长和滑动步长，而是对于每次滑动4/5窗长后重叠的1/5窗口长度部分进行处理。这部分经过两次静态色度标定，得到两段色标值**V\_first**、**V\_second**，对于这两段色标值分别乘上一个权值**W\_first**、**W\_second**，然后相加得到重叠部分最终的色标值。权值**W\_first**、**W\_second**需要满足以下两个条件：

1. **W\_first**随下标由1逐渐降低到0，**W\_second**随下标由0逐渐上升到1
2. **W\_first**、**W\_second**相同下标值相加等于零

$$V_{result} = \vec{w}_{first}^T V_{first} + \vec{w}_{second}^T V_{second}$$

实际上一般使用线性渐变权值，通过渐变过渡法便可以有效消除边界的阶跃断层。



### 3.自适应动态色度标定

#### API

##### 自适应法的动态色度标定

```
def dynamicOperation_AutoAdjust(data:pd.DataFrame,function, winlen:int = 300)
```

- 输入参数

变量名	type	说明
data	pandas.DataFrame	index为深度值，标签为“DEPTH”；每一列为不同方位的数据，列标签不做要
function	callable(函数)	处理每一个数据窗数据的函数（如 <code>histogramEqualization(data:pd.DataFrame)</code> ）
winlen	int	窗长

- 返回值
  - **type:** pandas.DataFrame
  - index为深度值，标签为“DEPTH”；每一列为不同方位的数据，列标签与输入值相同

### 自适应法的动态色度标定\_快速

```
def dynamicOperation_QuicklyAutoAdjust(data:pd.DataFrame,transform_function,
winlen:int = 300)
```

- 输入参数

变量名	type	说明
data	pandas.DataFrame	index为深度值，标签为“DEPTH”；每一列为不同方位的数据，列标签不做要
transform_function	callable(函数)	要求接收一个参数data，返回处理每一个深度点数据的函数（如 <code>transform_HE(data:pd.DataFrame)</code> ）
winlen	int	窗长

- 返回值
  - **type:** pandas.DataFrame
  - index为深度值，标签为“DEPTH”；每一列为不同方位的数据，列标签与输入值相同

### 函数说明

参考计算视觉的自适应直方图均衡方法，对动态色度标定进行改进得到自适应动态色度标定。自适应动态色度标定同样需要选定一个窗长，对于每个深度点，截取前后相当于1/2窗长的数据（总长度为一个窗长），从这段数据中得到一个静态色度标定映射函数，这个映射函数只会作用选取的深度点，随后选取下一个深度点重复操作。这样由于测井数据不会产生类似动态色度标定时在两个相邻窗上的突变，映射函数的参数S与V\_offset会随着深度点渐变，这样也可以有效的消除边界的阶跃断层。

## 五、结论

通过几种对于动态色度标定的改进方法，可以有效的消减和消除由于相邻窗上色度标定映射函数的参数差异过大导致的阶跃断层，测井解释人员可以更好的从动态色度标定的方位成像图中判断地层、裂隙等信息。



