In [10]:
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_absolute_error
```

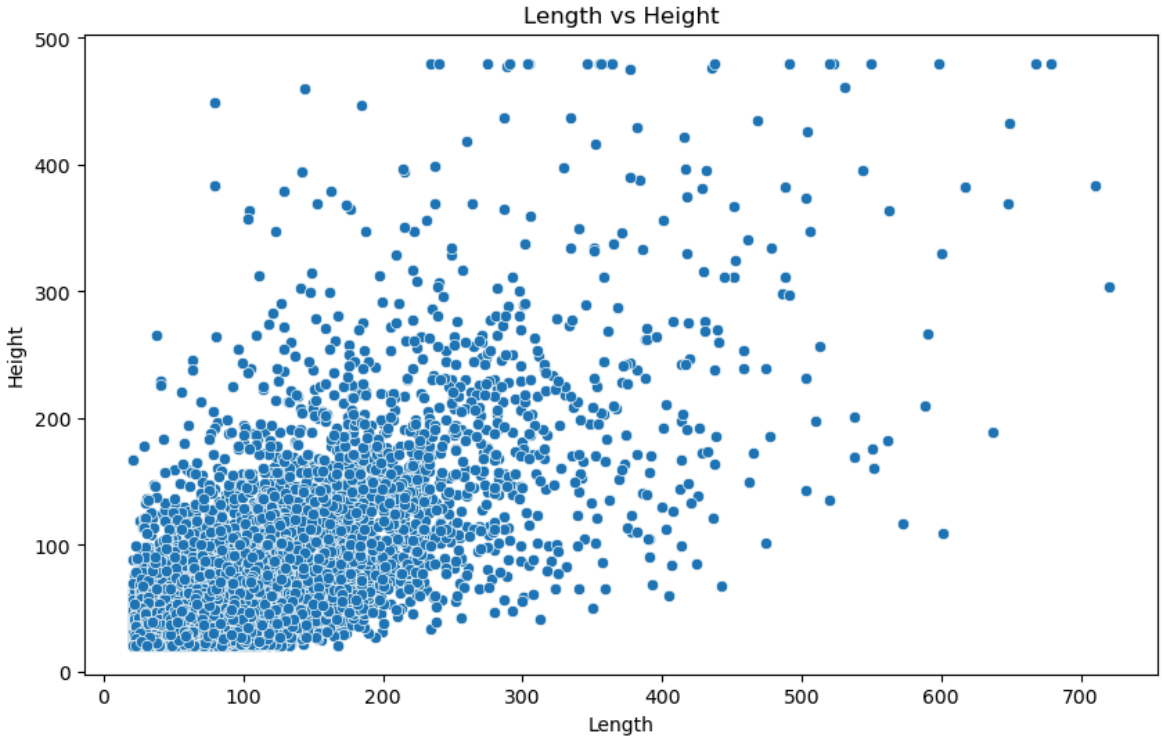
In [11]:
```
# Load dataset
file_path = r"C:\Users\Godfather Haiku\Desktop\robotics.csv"
data = pd.read_csv(file_path)
```

In [12]:
```
# Check for missing values
print("Missing values:\n", data.isnull().sum())
```

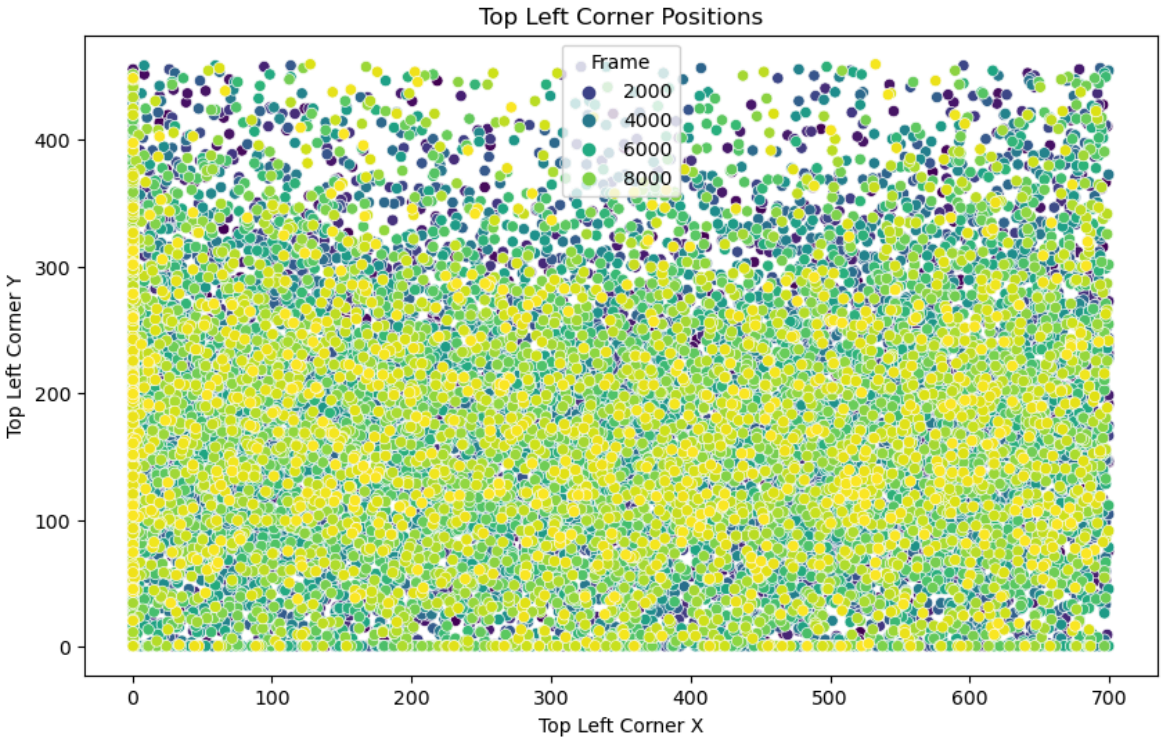
```
Missing values:
 Frame             0
TopLeftCornerX    0
TopLeftCornerY    0
Length            0
Height            0
dtype: int64
```

In [13]:
```
# Descriptive statistics
print(data.describe())

# Visualize Length vs Height
plt.figure(figsize=(10, 6))
sns.scatterplot(data=data, x='Length', y='Height')
plt.title('Length vs Height')
plt.xlabel('Length')
plt.ylabel('Height')
plt.show()
```

```
              Frame  TopLeftCornerX  TopLeftCornerY        Length  \
count  18867.000000    18867.000000    18867.000000  18867.000000
mean    4865.841469      321.963720      172.409843     71.715429
std     2821.505307      217.959299      101.717526     60.326411
min        1.000000        0.500000        0.500000     21.000000
25%     2430.000000      122.500000       95.500000     36.000000
50%     4881.000000      317.500000      164.500000     53.000000
75%     7281.000000      515.500000      239.500000     84.000000
max     9766.000000      699.500000      459.500000    720.000000

             Height
count  18867.000000
mean      49.871469
std       45.781198
min       21.000000
25%       26.000000
50%       34.000000
75%       53.000000
max      480.000000
```

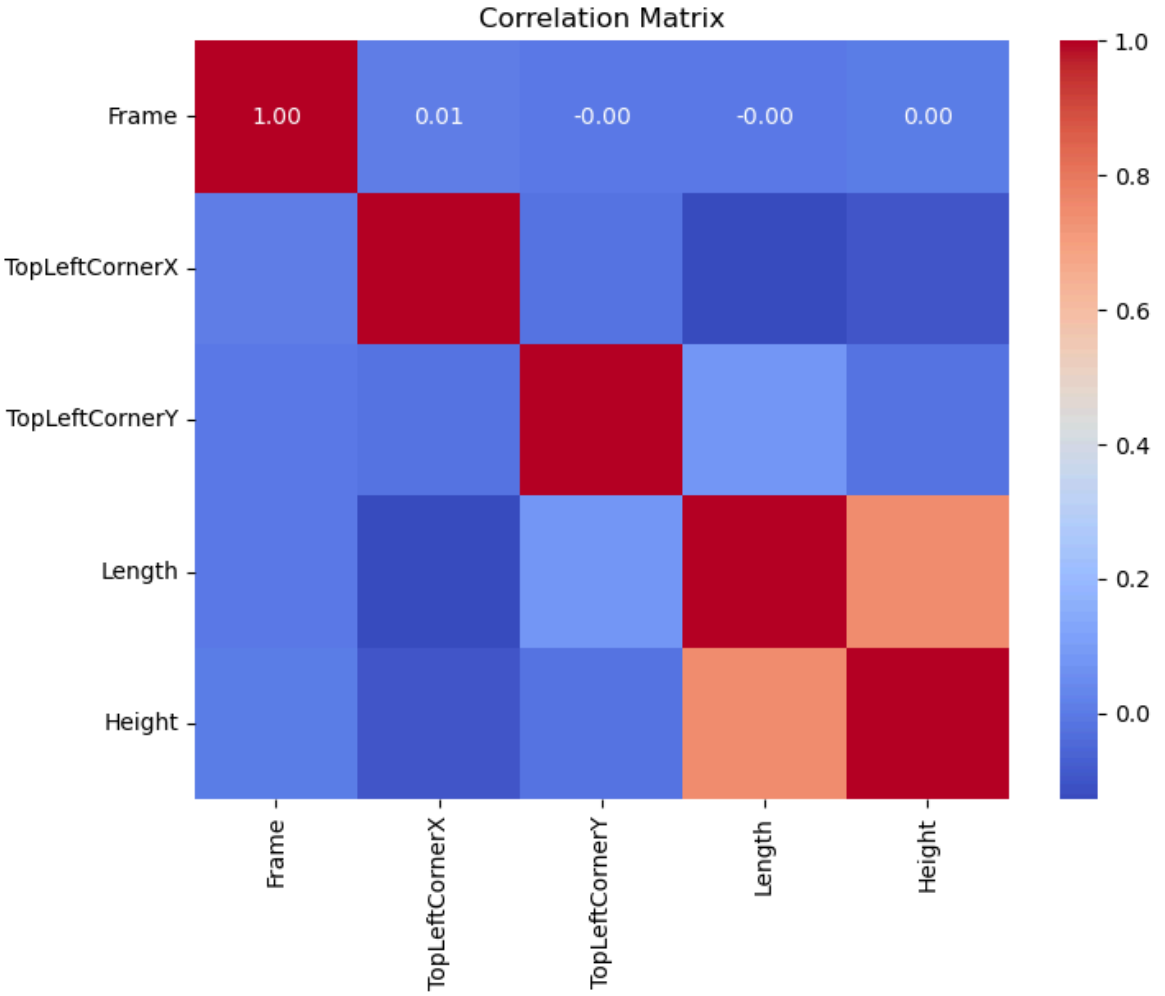


In [14]:
```
# Visualize TopLeftCorner Positions
plt.figure(figsize=(10, 6))
sns.scatterplot(data=data, x='TopLeftCornerX', y='TopLeftCornerY', hue='Frame',
plt.title('Top Left Corner Positions')
plt.xlabel('Top Left Corner X')
plt.ylabel('Top Left Corner Y')
plt.show()
```



In [15]:
```
# Correlation matrix
correlation_matrix = data.corr()
plt.figure(figsize=(8, 6))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt='.2f')
```

```
plt.title('Correlation Matrix')
plt.show()
```



In [16]:
```
# Define features and target variable
X = data[['Length', 'TopLeftCornerX', 'TopLeftCornerY']]
y = data['Height']

# Train-test split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_
```

In [17]:
```
# Model training
model = LinearRegression()
model.fit(X_train, y_train)

# Predictions
predictions = model.predict(X_test)
```

In [18]:
```
# Evaluation
mae = mean_absolute_error(y_test, predictions)
print(f'Mean Absolute Error: {mae}')
```

```
Mean Absolute Error: 16.942916786582973
```