A Hybrid Clustering Pipeline for

Mining Local Patterns in 3D Point Cloud

**Abstract:**

**Introduction:** With recent advances in high-resolution 3D scanning technology and extensive use of LIDAR imaging, 3D point cloud data are now available for many applications. 3D data represent true geometry of real-world objects, which is often compromised in conventional texture of 2D imaging and even in Kinect-based depth sensing. Despite richness in data, 3D point clouds are unstructured and unordered with inconsistencies in point counts, which turns machine learning (ML) of such data structure challenging. Existing literature either maps 3D geometric features (e.g., depth, curvature) onto 2D grid texture~\cite{} or creates stacks of 2D image slices from 3D model~\cite{} to leverage traditional image recognition pipelines. Such arbitrary 3D to 2D data projections for technical convenience may inadvertently discard useful information of true geometry in 3D data~\cite{Frenet}. Therefore, mining effective patterns directly from 3D point clouds, without introducing lossy 3D-2D data projections, is a nontrivial task.

The task of mining patterns in 3D data has been investigated in objects~\cite{} and human face recognition~\cite{} applications. The analysis of the human face in 3D is an active area of research that facilitates several emerging areas of research, including computational behavioral sciences~\cite{JOLT paper, 3DASD}, biometrics and affective computing~\cite{}, and forensics and medicine~\cite{}. A majority of these applications mine holistic patterns or global shape in 3D to differentiate between a ‘car’ and a ‘table’ or between the face of ‘John’ from that of ‘David’. In contrast, a more challenging task is to recognize local deformation patterns in 3D facial data that differentiate a surprised expression from an angry expression. The task of learning local patterns has been previously addressed by extracting and analyzing changes in local geometric and shape features. However, many of these analyses require grid-based representation of the 3D face that is an approximation of the actual 3D geometry. However, the mining of local patterns directly from raw 3D facial point cloud data is not well explored in the literature. In this paper, we propose to learn local patterns directly from raw 3D point cloud by addressing the following three challenges.

First, 3D point clouds are inconsistent in resolution and the number of points vary from one cloud to another. A common strategy is to sample and interpolate 3D point cloud on to a uniform 2D grid that facilitates operations similar to pixel-based processing. Instead, we propose a distance-based clustering method as a means to directly segment the facial point cloud into distinct and meaningful regions. Second, clustering algorithms often yield different labels for the same region of the point cloud that depends on initial enumeration of the seeds. This makes tracking and identifying the same spatial region with the same label across different point clouds challenging. In this paper, labels for different segments are consistently identified across all face samples to facilitate local pattern classification. Local features mined from these segments are compared across faces for classifying seven prototypical facial expression patterns. Several clustering algorithms have been tested to evaluate facial segmentations. Third, to achieve a constant dimensional feature space for point clouds of varying size, we use parameters after learning a mixture of Gaussian distributions using the expectation maximization algorithm. These parametric features are used to identify most discriminative facial segments for different expression patterns.

**Methods:**

We propose a hybrid of clustering algorithms for learning of local geometric patterns from 3D point clouds of the human face. The pipeline is shown in Figure~\ref{fig:01} and further discussed below.

**Data set:** The proposed pipeline for analyzing 3D point clouds is evaluated using the Binghamton University 3D facial expressions (BU-3DFE) data set~\cite{}. The data set has 3D faces with seven benchmarked facial expressions. The seven prototypical expressions are: happiness, anger, fear, sadness, disgust, surprise, and neutral. The 3D facial data are acquired using 3dMD – a commercially available high-resolution 3D facial imaging system~\cite{}. Additionally, the data set provides well cropped 3D faces that are free of outlier points. We extract and subsequently analyze 3D point coordinates (x, y, z) of the provided 3D models in WRL (a virtual reality modeling language extension) format. The class labels provided with the data set are used to perform supervised classification of seven facial expressions.

**Segmentation of 3D point clouds:** Image segmentation is an active area of research that studies similarity among group of pixel intensities or texture features. In contrast to 2D imaging, 3D point clouds form a surface topology with varying point resolution and count. Assuming the symmetric shape of the human face, we segment eight distinct facial regions. First, we employ and evaluate different clustering algorithms and distance metrics to group the 3D facial point clouds into eight facial segments. For facial point cloud segmentation, we compare the performances of K-means clustering (with Euclidean and Mahalanobis distances), spectral clustering, and Gaussian mixture model. The algorithm that yields most meaningful facial segments is chosen for the subsequent local pattern analysis.

**Alignment of 3D point clouds: The** registration of 3D point clouds is important, especially when raw 3D coordinate data are used for the pattern analysis. We perform the registration at two levels: 1) the nose tip is identified as the highest Z-axis coordinate that is used to align and adjust all faces such that the nose tip is at the origin of the 3D Cartesian coordinate and 2) after segmentation of the facial patches, all patches are aligned to a common patch centroid.

**Parameter learning for feature extraction** The eight 3D facial patches obtained from the previous step are further grouped into Q clusters by learning Q 3D Gaussian distributions. The Gaussian distribution parameters are learned via the expectation maximization (EM) algorithm. The EM algorithm learns a set of Gaussian parameter vectors: S = {w, mu, sigma} for each cluster distribution that maximizes the likelihood function P(xj|ci) = exp {}, which is a similarly measure between the input 3D point coordinate xi and the cluster centroid $\mu\_j$. The soft cluster assignment of each 3D coordinate point x is determined by the posterior probability P(c\_i|x\_j) of the cluster c\_i given the input x\_j as below.

where, x is 3D Cartesian coordinate, ci represents one of the 8 clusters and, w\in\Re, \mu\in\Re^3, \sigma\in\Re^{3x3} are weight, mean vector and covariance matrix of each Gaussian distribution. A sample point $x\_i$ belongs to the $c\_j$ cluster that yields the maximum P(c\_i|x\_j).

There are Q weight parameters (w) for Q Gaussian clusters. The 3D mean vector (\mu) for each Gaussian cluster yields a total of 3XQ parameter values. We extract the 3D diagonal vector from the 3x3 covariance matrix ($\sigma$) of each Gaussian cluster. Total covariance parameters for each patch are 3XQ. Each patch yields a total of (3Q + 3Q + Q) or 7Q parameters. Thus, individual 3D facial point cloud is represented by 8\*Q\*7 or 56\*Q features.

**Importance of features and facial segments:** The importance of individual 56\*Q features is interpretable since each feature has a physical significance and can be related to a spatial location of the point cloud. A naïve and unsupervised way to look at the feature importance is to measure individual feature variances across seven facial expressions of the same human subject. Features with low or zero variability suggest that those are not sensitive to local variations in the face. Features with highest variances across facial expressions can be used to identify their corresponding facial segments that are most sensitive to facial expressions.

**Classification and model evaluation:** We perform classification of seven-class facial expressions using one-versus-all scheme. A nested10x2 fold cross-validation (CV) is proposed for the model evaluation where a two-fold inner CV is nested within each of ten-fold CV. The two-fold CV split the nine-fold training data 50%-50% to identify the best classifier hyperparameters via grid search. The best model hyperparameters are then used to train the classifier model on the nine-fold data to test on the left-out data fold. [Classifier models] A mean area under the receiver operating characteristics curve (AUC) is used to evaluate the overall classification performance since AUC accounts for both model sensitivity and specificity at various classification scores. The classification performance reveals 1) how good is the point cloud clustering and 2) how informative are the parametric features in representing local patterns in point clouds.

**Results:**

The proposed pipeline is evaluated using 50 3D point cloud samples per facial expressions with a total of 350 samples for seven expression labels. The key results are discussed below.

**Facial segmentations:** To be consistent with the cluster labels and locations across 3D facial point clouds, we use a reference face to first obtain a centroid using K-means clustering algorithm. The centroid is then used as initial seed prior to segmenting all other facial point clouds. We apply four clustering algorithms: 1) K-means Euclidean, 2) K-means Mahalanobis, 3) Gaussian mixture model, 4) spectral clustering with K-means to visualize how these algorithms effectively segment different facial regions. Figure~\ref{fig:02} shows representative results of facial segmentations. The K-means algorithm identifies most meaningful facial parts with uniform cluster segments as shown in respective Silhouette plots. Therefore, we use K-means algorithm for facial segmentation step in the proposed pipeline.

**Feature extraction and interpretation:** Following the segmentation of individual faces into eight patches, we further group each patch into three clusters using the Gaussian Mixture Model (GMM). The three clusters from each of the eight patches yield a total of 24 segments per 3D facial point cloud. Each of the 24 segments is modeled by learning seven Gaussian parameters, which yield a total of 168 features per 3D point cloud. The variances in 21 features across 7 facial expressions are shown in Table~\ref{tab:01} corresponding to eight distinct facial patches. The Table shows the most varying patches and parameters across seven facial expressions.

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | GMM cluster 1 | | | GMM Cluster 2 | | | GMM cluster 3 | | |  |
| Patch Number | Facial Region | w  (Scalar) | Mu  (vector) | sigma | w | mu | sigma | w | mu | sigma | Best |
| 1 | Nose | 0.0024567162 | 353.43207957 | 5652.7968 | 0.00326004 | 380.01227 | 5643.4404 | 0.00307549 | 377.6539 | 7244.34429 | GMM-3, mu |
| 2 | Lower right cheek | 0.0214158764 | 327.87430248 | 11690.6093 | 0.02051143 | 346.68748 | 12198.8687 | 0.02121530 | 352.1070 | 11365.1523 |  |
| 3 | Left eye with forehead | 0.0029500299 | 313.720693 | 13283.3979 | 0.00315426 | 376.60954 | 12124.2247 | 0.00326673 | 345.7653 | 13772.5892 |  |
| 4 | Right eye with forehead | 0.0139694297 | 371.947477 | 8934.8977 | 0.01437106 | 393.23523 | 9042.5672 | 0.01677717 | 386.2695 | 9528.31204 |  |
| 5 | Lower left cheek | 0.0032656583 | 378.986724 | 6258.9564 | 0.00290010 | 394.6490 | 6100.73029 | 0.00370627 | 388.2482 | 6701.9553 |  |
| 6 | Chin | 0.0141713027 | 383.883203 | 10051.8877 | 0.01124536 | 373.87821 | 8150.1237 | 0.01443702 | 373.6247 | 9835.02779 |  |
| 7 | Upper left cheek | 0.0022738467 | 338.945512 | 14599.4937 | 0.00246808 | 353.07448 | 12827.9386 | 0.00244417 | 342.7935 | 14059.1117 |  |
| 8 | Upper right cheek | 0.0128280681 | 367.948500 | 8760.4762 | 0.01106012 | 384.91348 | 8071.0689 | 0.01414595 | 353.2775 | 8519.4813 |  |
| Best Patch |  | 2 | 6 | 7 | 2 | 5 | 7 | 2 | 5 | 7 |  |

Table 1. Patch 1: nose region, Patch

mu = [4, 5, 6]

3X 5 (1+2+2)

3X7

var (mu) = var (mu[0]) + var (mu[1]) + var(mu[2])

var (sigma) = var (sigma[0]) + var (sigma[1]) + var(sigma[2])

21 x 8 = 168

21 ((1+3+3)X 3 GMM clusters)

Var(w) = Summation\_{i =1 }^{i=7} (w - mean (w))

Xx = mu1+mu2^2+mu3^2

**Classification performance:** In seven-class classification, we have achieved a mean AUC of 0.72 and the confusion matrices are shown in Figure~\ref{fig:03}.

**Conclusions:**

\begin{algorithm}

\vspace{5pt}

\caption{Local feature extraction via hybrid clustering}

\begin{algorithmic}

\STATE Input: 3D point cloud, A = $[a]\_{mx3}$

\STATE Output: n-dimensional local feature vector, fvect

\item Select top k word in matrix, $W\_k = [x]\_{kx1}$, (k$\leq$m).

\STATE A $\leftarrow$ A - A [A [z] == max(A[z])]

\STATE patches $\leftarrow$ cluster (A, $n\_patch$ = 8)

\FOR {t =1 $\rightarrow$ n\_patch}

\STATE patch = patch - centroid (patch)

\STATE w [t] = gmm (patch, n\_cluster = 3)

\STATE \mu [:, t] = gmm (patch, n\_cluster = 3)

\STATE \cov\_matrix = gmm (patch, n\_cluster = 3)

\STATE \sigma[:, t] = diag (cov\_matrix)

\STATE fvect = [w[t], flatten (\mu[:,t]), flatten(\sigma[:,t])]

\ENDFOR

\FOR {t =1 $\rightarrow$ n\_patch}

\FOR {r =1 $\rightarrow$ n\_cluster = 3}

{w, \mu, \sigma} $\rightarrow$ gmm\_cluster [r, patch[t]]

wvar [t, r] = \sum\_{q=1}^{q = n\_class} \frac{[w[q] - \bar(w)]^2}{n\_class}

\sigma^2 ($w\_r$ [t])

\STATE Repeat below separately for

\STATE Svar and SL2 in place of S

\FOR {r = 1 $\rightarrow$ m}

\STATE index [r] = argmax (S)

\STATE maxValue = S [argmax (S)]

\STATE S$\leftarrow$ remove maxValue from S

\ENDFOR

\STATE Wk $\leftarrow$ W [index [1:k]]

\end{algorithmic}

\end{algorithm}