Практическое занятие 2.2 Создание новых материалов. Тканевая инженерия. Компенсационные материалы

21 февраля 2022 г.

Содержание

1	Введение	1
2	Цель	1
3	Теория пленок Ленгмюра-Блоджетта 3.1 Обзор	6 4 6 4
4	Модель	4
5	Прыжтическая работа 5.1 Подготовка окружения выполнения	4
6	Заключение	18

1 Введение

Лабораторная работа по теме 2.2 "Биоматериалы и биомеханика ткани"курса М1.Б.4 "Современные проблемы биомедицинской и экологической инженерии"для подготовки магистров по направлению 12.04.04 "Биотехнические системы и технологии".

На лекциях были рассмотрены Ленгмюровские пленки.

Практическое занятие позволяет на практике освоить методы компьютерного зрения при работе с Ленгмюровскими пленками.

2 Цель

Получить навыки применения практических данных в построении графического материала для визуализации научных данных и результатов расчетов для углубления компетенции представлять современную научную картину мира, выявлять естественнонаучную сущность проблемы, формулировать задачи, определять пути их решения и оценивать эффективность

выбора и методов правовой защиты результатов интеллектуальной деятельности с учетом исследований, разработки и проектирования биотехнических систем и технологий.

3 Теория пленок Ленгмюра-Блоджетта

3.1 Обзор

В 1980-х годах был большой энтузиазм по поводу молекулярных сборок на основе пленок Ленгмюра-Блоджетт (LB). Были предложены концепции для следующего тысячелетия, такие как «Молекулярная электроника, в которой органические молекулы выполняют активную функцию в обработке информации, а также в передаче и хранении» [3, 4, 5]. Эти предложения вызвали много ожиданий.

На текущий момент пленки LB служат в основном модельными поверхностями для изучения взаимодействий.

Как было показано в лекции,в исследованиях и производстве биоматериалов применяются, как обычные пленки LB, так и новые типы, полученные путем организации наночастиц на границе воздух-вода.

Исчерпывающую информацию по LB можно найти в [2], далее будут приведены лишь выборочные материалы из этой публикации.

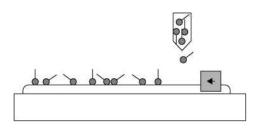
3.2 Что делает пленки LB привлекательными?

Привлекательными свойствами пленок Ленгмюра-Блоджетт являются внутренний контроль структуры внутреннего слоя вплоть до молекулярного уровня и точный контроль толщины полученной пленки.

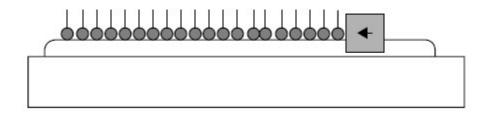
Сложные желоба LB позволяют нам обрабатывать несколько материалов с различными функциональными возможностями и дают возможность настраивать архитектуру слоев в соответствии с требованиями желаемых молекулярно-инженерных органических тонкопленочных устройств.

Более подробную информацию можно найти в книгах Гейнса [6] или Ульмана [7]; последний также представляет собой хорошее введение в инструменты анализа поверхности, которые обычно используются для исследования структуры монослоев и различных физических свойств. Первым этапом процесса осаждения LB является формирование четко определенного монослоя на границе воздух-вода. Эти так называемые ленгиюровские монослои являются пленкой-предшественником в производстве LB. Процесс подготовки схематично показан на следующей поэтапной схеме процесса осаждения Ленгиюра-Блогетта.

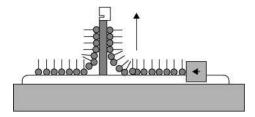
Амфифил растворяют в органическом растворителе и затем распределяют на границе воздухвода. Растворитель испаряется и остается монослой амфифила на границе воздух-вода. Монослоем на границе воздух-вода можно дополнительно управлять с помощью подвижного барьера, позволяющего контролировать площадь на молекулу.



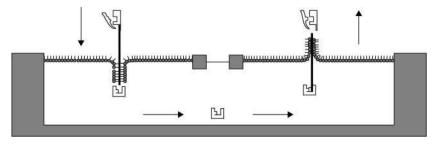
Монослоем на границе воздух-вода можно дополнительно управлять с помощью подвижного барьера, позволяющего контролировать площадь на молекулу



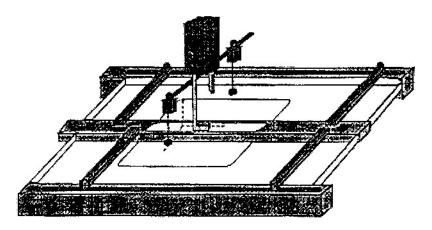
Монослой Ленгмюра можно переносить движением вверх на гидрофильную поверхность



и движением вниз на гидрофобную поверхность. Ванночка с двумя отсеками позволяет одновременно обрабатывать два разных материала



а запрограммированная последовательность погружения позволяет определять архитектуру слоя на молекулярном уровне.



Монослои на границе воздух-вода были тщательно изучены и обладают богатством фаз и структур. Они служат квазидвумерными модельными системами и, таким образом, привлекли значительное внимание исследователей.

Появление сложных инструментов анализа поверхности, таких как методы отражения и рассеяния рентгеновских лучей, вместе с новыми оптическими методами, такими как флуоресцентная микроскопия и микроскопия под углом Брюстера (ВАМ), предоставили подробную картину общих фазовых диаграмм, структуры и морфологии.

Рентгеновская дифракция скользящего падения выявила существование нескольких фаз, в которых алифатическая цепь наклонена относительно нормали к поверхности и в которых азимут наклона принимает четко определенное расположение относительно лежащего в основе порядка ориентации связи [8].

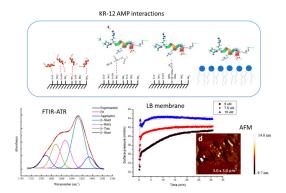
Организация азимута наклона может распространяться на макроскопические размеры и, таким образом, проявляется на изображениях ВАМ в различных аспектах, таких как образование доменов с внутренней структурой.

Обзорные статьи Mohwald [9], McConnel [10], Knobler and Desai [11], Riviere et al [12] и Knobler and Schwartz [13] являются прекрасными проводниками по огромному количеству публикаций, посвященных монослоям Ленгмюра.

4 Модель

Модель расчета повторяет эксперимент по исследованию физического поведения пептида KR-12 липидных пленках Ленгмюра-Блоджетта при его антимикробном действии против S.aureus и демонстрирует алгоритм определения толщины с помощью компьютерного зрения. [1]

В статье были представлены результаты, приведенные на следующем рисунке:



Для проведения практической работы воспользуемся лабораторными фотографиями опубливанных там же [1].

5 Практическая работа

5.1 Подготовка окружения выполнения

Для успешного выполнения кода на своем компьютере необходимо выполнить ряд подготовительных мероприятий.

5.1.1 Установка Anaconda

Anaconda - дистрибутив языков программирования Python и R, включающий набор популярных свободных библиотек, объединённых проблематиками науки о данных и машинного обучения.

Для скачивания пакета Anaconda надо перейти по ссылке и скачать дистрибутив для своей операционной системы:

https://www.anaconda.com/products/individual





Individual Edition

Your data science toolkit

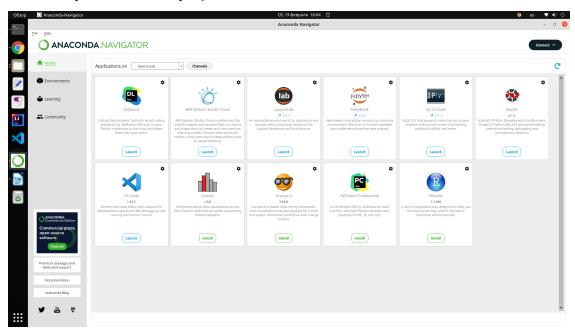
With over 25 million users worldwide, the open-source Individual Edition (Distribution) is the easiest way to perform Python/R data science and machine learning on a single machine. Developed for solo practitioners, it is the toolkit that equips you to work with thousands of open-source packages and libraries.



После установки необходимо запустить навигатор. В Windows и MacOS X его можно найти в списке приложений, в Linux его можно запустить из консоли

./anaconda-navigator

Окно навигатора показано на рисунке ниже.



5.1.2 Установка зависимостей

После установки Anaconda нам надо установить дополнительные библиотеки, для этого в консоли необходимо выполнить следующие команды. Их выполнение может занять значительное время, поэтому не стоит их прерывать из-за предположения, что они зависли. Если у вас возникнут сложности, поскольку со временем что то может измениться, то всегда можно обратиться с поиском в интернет и "загуглить" ошибку и способ ее устранения.

```
conda update --all --yes
conda install -c conda-forge google-colab
conda install -c conda-forge opencv
conda install -c conda-forge pandoc
conda install -c conda-forge nbconvert
conda install -c conda-forge pyppeteer
```

5.2 Пошаговая инструкция для создания блокнота в Jupyter Notebook

```
[51]: import numpy as np import matplotlib.pyplot as plt import pandas as pd import cv2 #Библиотека компьютерного зрения OpenCV
```

^{**} Изображение из https://doi.org/10.1016/j.bbamem.2021.183779 **

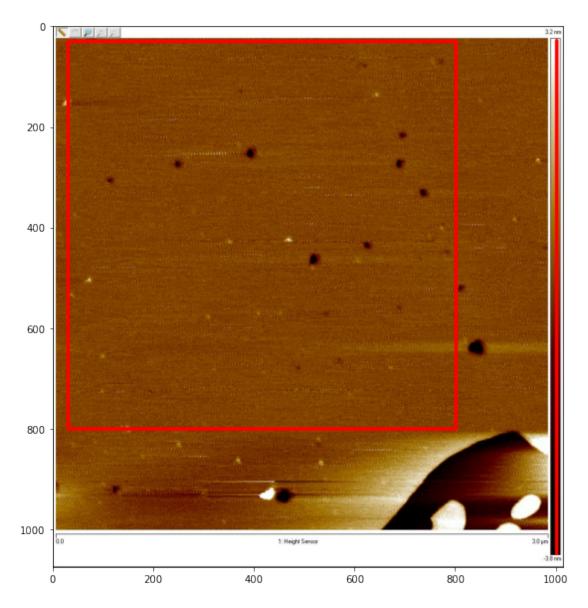
```
[52]: #Загружаем изображение

img = cv2.imread('images/PGPGCL_H2O_004.jpg')
img = cv2.cvtColor(img,cv2.COLOR_BGR2RGB)

#Обрезаем изображение

cv2.rectangle(img,(30,30),(800,800),(255,0,0),5); #Красный квадрам
cv2.rectangle(img,(1000,28),(1000,1048),(255,0,0),5); #Красная линия
plt.figure(figsize=(10,10))
plt.imshow(img)
```

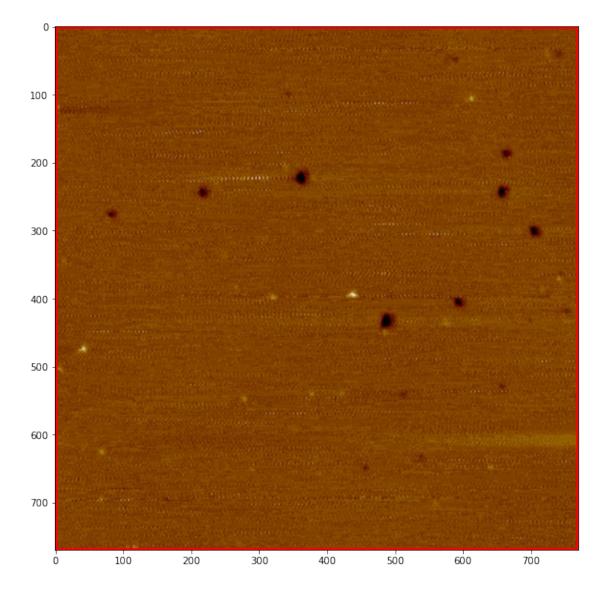
[52]: <matplotlib.image.AxesImage at 0x7fd61a89ff70>



```
[53]: #Переименуем красный квадрат и красную линию

bilayerImg = img[30:800,30:800] #Красный квадрат (двухслойный/bilayer)
scaleImg = img[28:1048,1000] #Красная линия (шкала высот)
plt.figure(figsize=(10,10))
plt.imshow(bilayerImg)
```

[53]: <matplotlib.image.AxesImage at 0x7fd61aac7b50>



[54]: bilayerImg.shape #Получить параметры красного квадрата в пикселях

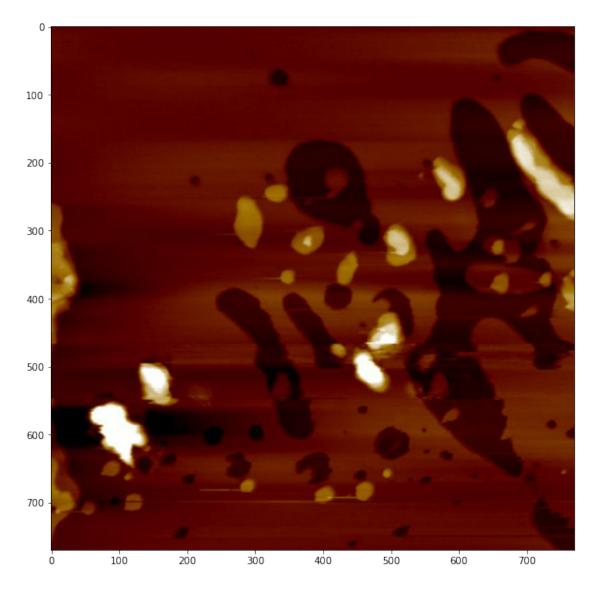
```
[54]: (770, 770, 3)
[55]: #Средний цвет RGB красного квадрата
      promRGB = np.mean(bilayerImg, axis=(0,1))
      promRGB = [int(i) for i in promRGB]
      promRGB
[55]: [132, 66, 2]
[56]: #Проверьте значения красной линии для і в scaleImg:
      for i in scaleImg:
          print(i)
     [255
            0
                0]
     [255
            0
                0]
[57]: | #Сравните цвет RGB каждого пикселя в красном квадрате с цветом RGB каждогоц
       →пикселя в красной линии.
      c = 0
      for i in scaleImg:
          c += 1
          if np.sum(abs(i - np.array(promRGB))) < 7:</pre>
              print('Match',c)
                                                       #Сопоставьте с более похожим
       →цветом по шкале высоты
     467 индекс средней высоты
[58]: #Средняя высота бислоя
      Bilayer_h = 3.2 + 3.8 #Высота микрофотографии АСМ
      Match = 467
                           #Индекс средней высоты
      Scale_pixls = 1020 #Количество пикселей в красной линии
      Top_h = 3.2
                            #Самая высокая точка на микрофотографии АСМ
      h = (Match*Bilayer_h)/Scale_pixls
      Real_h = Top_h-h
      print(Real_h)
     -0.004901960784313708
     В среднем липидный бислой был однородным.
     Изображение из https://doi.org/10.1016/j.bbamem.2021.183779
[59]: #Загружаем изображение
      img2 = cv2.imread('images/PGPGCL_H20_AMP_007_KR.jpg')
```

```
img2 = cv2.cvtColor(img2,cv2.COLOR_BGR2RGB)

#Обрезаем изображение

bilayerImg = img2[30:800,30:800]
scaleImg = img2[28:1048,1000]
plt.figure(figsize=(10,10))
plt.imshow(bilayerImg)
```

[59]: <matplotlib.image.AxesImage at 0x7fd61a7e4460>



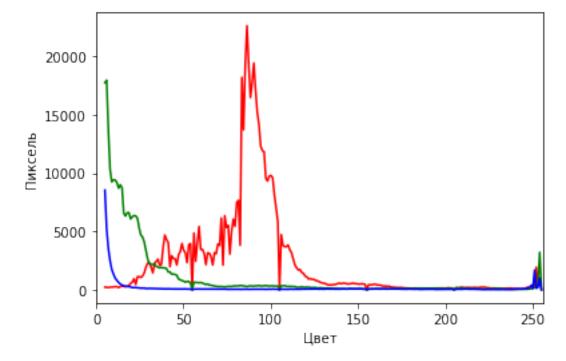
[60]: # Создайте график гистограммы с тремя линиями, по одной для каждого цвета.

```
colors = ("r", "g", "b")
channel_ids = (0, 1, 2)

plt.xlim([0, 256])
for channel_id, c in zip(channel_ids, colors):
    histogram, bin_edges = np.histogram(
        bilayerImg[:, :, channel_id], bins=256, range=(5, 256)
    )
    plt.plot(bin_edges[0:-1], histogram, color=c)

plt.xlabel("Цвет")
plt.ylabel("Пиксель")

plt.show()
```



```
[61]: #Средний цвет RGB бислоя после 30 минут взаимодействия с KP-12 AMP

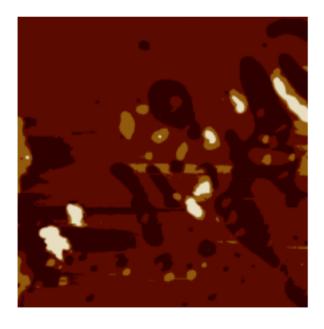
promRGB = np.mean(bilayerImg, axis=(0,1))
promRGB = [int(i) for i in promRGB]
promRGB
```

[61]: [86, 16, 5]

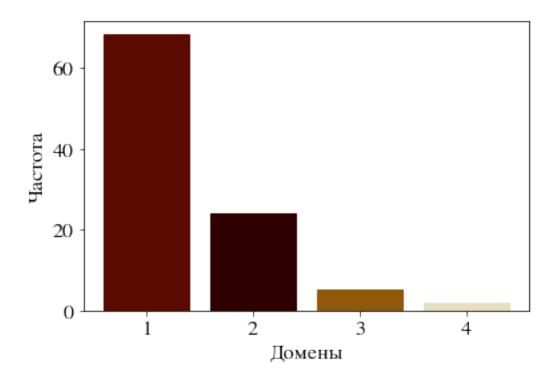
```
[62]: \#Cравните цвет RGB каждого пикселя в двухслойном изображении с цветом RGB_{\sqcup}
       → каждого пикселя на шкале высоты.
      c = 0
      meann = []
      for i in scaleImg:
          c += 1
          if np.sum(abs(i - np.array(promRGB))) < 22:</pre>
            print('Match',c)
                                                         #Сопоставьте с более похожим_
       →цветом на шкале высот
            meann.append(c)
     Match 581
     Match 582
     Match 583
     Match 584
     Match 585
     Match 586
     Match 587
     Match 588
     Match 589
     Match 590
     Match 591
     Match 592
     Match 593
     Match 594
     Match 595
     Match 596
     Match 632
     Match 633
     Match 634
     Match 635
[63]: np.mean(meann) #Индекс средней высоты
[63]: 597.5
[64]: #Средняя высота бислоя через 30 мин взаимодействия с пептидом
      Bilayer_h = 14.6 + 9.7 #Высота микрофотографии АСМ
      Match = 597
                              #Индекс средней высоты
      Scale_pixls = 1020
                            #Количество пикселей в красной линии
      Top_h = 14.6
                              #Самая высокая точка на микрофотографии АСМ
      h = (Match*Bilayer_h)/Scale_pixls
      Real_h = Top_h-h
      print(Real_h)
```

0.377352941176472

```
[65]: #Процесс сегментации
      pixel_values = bilayerImg.reshape((-1, 3)) #Изменить изображение
      pixel_values = np.float32(pixel_values) #Koheepmupyem & float
      #pixel_values.shape
      #pixel_values
      #Критерий сегментации
      criteria = (cv2.TERM_CRITERIA_EPS + cv2.TERM_CRITERIA_MAX_ITER, 100, 0.2)
      _, labels, (centers) = cv2.kmeans(pixel_values, k, None, criteria, 10, cv2.
      →KMEANS_RANDOM_CENTERS)
      #Преобразование обратно в 8-битные значения
      centers = np.uint8(centers)
      #Сгладить массив меток
      labels = labels.flatten()
      centers #Средний цвет RGB каждого кластера
[65]: array([[ 91, 10,
                        0],
             [144, 89, 11],
             [46, 0, 0],
             [231, 221, 192]], dtype=uint8)
[66]: #labels
      #labels.shape
      segmented_image = centers[labels.flatten()]
      #segmented_image
      #Изменить форму до исходного размера изображения
      segmented_image = segmented_image.reshape(bilayerImg.shape)
      #Выводим изображение
      fig = plt.figure()
      plt.imshow(segmented_image)
      plt.axis('off')
      fig.savefig('supp_Fig_1aa.png',dpi =1200)
```



```
| #Yacmommbil zpaφuk δλα καπδοιο κλαcmepa | pd.Series(labels).value_counts()/labels.shape[0]*100 | propor = pd.Series(labels).value_counts()/labels.shape[0]*100 | plt.rcParams.update({'font.size': 16, 'font.family': 'STIXGeneral', 'mathtext. \( \to fontset': 'stix' \) | fig = plt.figure() | bar = plt.bar(['1','2','3','4'], propor.values) | bar[0].set_color((91/255,10/255,0)) | bar[1].set_color((46/255,0,0)) | bar[2].set_color((144/255,88/255,11/255)) | bar[3].set_color((231/255,221/255,191/255)) | plt.ylabel('Частота') | plt.xlabel('Домены') | fig.savefig('supp_Fig_1b.png',dpi = 1200) | plt.ylabel('supp_Fig_1b.png',dpi = 1200) | plt.ylabel('supp_Fig_1b.png',dpi
```



```
[68]: centers #Средний цвет RGB каждого кластера
[68]: array([[ 91,
                   10,
                          0],
             [144,
                   89,
                        11],
             [ 46,
                     Ο,
                          0],
             [231, 221, 192]], dtype=uint8)
[69]: #Самый яркий домен
      c = 0
      meann = []
      for i in scaleImg:
          c += 1
          if np.sum(abs(i - np.array(centers[0]))) < 9:</pre>
            print('Match',c)
            meann.append(c)
     Match 589
     Match 590
     Match 591
     Match 592
[70]: #Средняя высота самого яркого домена
      Bilayer_h = 14.6 + 9.7 #Bысота микрофотографии ACM
```

```
Match = 102
                        #Индекс средней высоты
Scale_pixls = 1020
                        #Количество пикселей в красной линии
Top_h = 14.6
                        #Самая высокая точка на микрофотографии АСМ
h = (Match*Bilayer_h)/Scale_pixls
Real_h = Top_h-h
print(Real_h)
```

12.17

```
[71]: #The bright domain
      c = 0
      meann = []
      for i in scaleImg:
          c += 1
          if np.sum(abs(i - np.array(centers[1]))) < 35:</pre>
            print('Match',c)
            meann.append(c)
```

Match 370 Match 371 Match 372 Match 373 Match 374 Match 379

Match 380

Match 401

[72]: #Средняя высота яркого домена

```
Bilayer_h = 14.6 + 9.7 #Высота микрофотографии АСМ
Match = 371
                       #Индекс средней высоты
Scale_pixls = 1020 #Количество пикселей в красной линии
Top_h = 14.6
                      #Самая высокая точка на микрофотографии АСМ
h = (Match*Bilayer_h)/Scale_pixls
Real_h = Top_h-h
print(Real_h)
```

5.761470588235294

```
[73]: #The dark domain
      c = 0
      meann = []
      for i in scaleImg:
          c += 1
          if np.sum(abs(i - np.array(centers[2]))) < 7:</pre>
```

```
print('Match',c)
            meann.append(c)
     Match 765
     Match 766
     Match 767
     Match 768
     Match 769
     Match 770
     Match 771
     Match 772
[74]: #Mean height of the dark domain
      Bilayer_h = 14.6 + 9.7 #Height of the AFM micrograph
      Match = 770
                              #Index of the mean height
                            #Amount of pixels in the red line
      Scale_pixls = 1020
      Top_h = 14.6
                              #The highest point in the AFM micrograph
      h = (Match*Bilayer_h)/Scale_pixls
      Real_h = Top_h-h
      print(Real_h)
     -3.74411764705882
[75]: #Фоновый домен
      c = 0
      meann = []
      for i in scaleImg:
         c += 1
          if np.sum(abs(i - np.array(centers[3]))) < 7:</pre>
            print('Match',c)
            meann.append(c)
     Match 104
     Match 105
     Match 106
     Match 107
     Match 108
[76]: #Средняя высота фонового домена
      Bilayer_h = 14.6 + 9.7 #Высота микрофотографии АСМ
      Match = 591
                              #Индекс средней высоты
      Scale_pixls = 1020
                            #Количество пикселей в красной линии
      Top_h = 14.6
                              #Самая высокая точка на микрофотографии АСМ
      h = (Match*Bilayer_h)/Scale_pixls
```

```
Real_h = Top_h-h
print(Real_h)

0.5202941176470617

[]:

[]:
```

6 Заключение

В результате практической работы был продемонстрирован на практике метод измерения толщины LB пленки с помощью библиотеки компьютерного зрения OpenCV.

Пленки LB были предложены для многих практических применений, охватывающих широкий спектр различных областей от биодатчиков до просветляющих покрытий и полностью оптических процессоров. Выли достигнуты значительные успехи в конструировании молекул и LB-структур с функциями, и взаимосвязь между макроскопическими и молекулярными структурами была рассмотрена очень подробно. Тем временем было достигнуто четкое понимание основных принципов, регулирующих упаковку и структуру пленки LB. Однако, несмотря на этот значительный прогресс, пленки LB еще не нашли своего выхода на рынок, а конкурентные подходы оказались более успешными с коммерческой точки зрения. Тем не менее, методы LB остаются потенциально полезными из-за своей простоты и способности создавать молекулярные сборки, определенные на молекулярном уровне. В этом отношении пленки LB будут продолжать служить ценными модельными системами для решения основных научных проблем, таких как смачиваемость, трение и молекулярное распознавание, поэтому, полученный практический опыт является актуальным и может быть применен в дальнейшей научной и производственной деятельности.

Список литературы

- [1] Katerine Chanci, Johnatan Diosa, Marco A. Giraldo, Monica Mesa, Physical behavior of KR-12 peptide on solid surfaces and Langmuir-Blodgett lipid films: Complementary approaches to its antimicrobial mode against S. aureus, Biochimica et Biophysica Acta (BBA) Biomembranes, Volume 1864, Issue 1,2022,183779,ISSN 0005-2736, https://doi.org/10.1016/j.bbamem. 2021.183779.
- [2] Handbook of Applied Surface and Colloid Chemistry. Edited by Krister Holmberg, 2001 John Wiley and Sons, Ltd
- [3] Roberts, G., Langmuir-Blodgett Films, Plenum Press, New York, 1990.
- [4] Roberts, G. G., An applied science perspective of Langmuir-Blodgett Films, Adv. Phys., 34, 475-512 (1985).
- [5] Peterson, I. R., Langmuir-Blodgett techniques, in The Molecular Electronic Handbook, Mahler, G., May, V. and Schreiber, M. (Eds), Marcel Dekker, New York, 1994, pp. 47-77.

- [6] Gaines, G. L., Insoluble Monolayers at Liquid-Gas Inter- faces, Wiley-Interscience, New York, 1966.
- [7] Ulmann, A., Introduction to Ultrathin Organic Films: From Langmuir-Blodgett to Self-Assembly, Academic Press, San Diego, CA, 1991.
- [8] Kaganer, M., Peterson, I. R., Kenn, R., Shih, M., Durbin, M. and Dutta, P., Tiltod phoses of fatty acid mono- layers, J. Chem. Phys., 102, 9412-9422 (1995).
- [9] Mohwald, H., Phospholipid and phospholipid-protein monolayers at the air/water interface, Annu. Rev. Phys. Chem., 41, 441-••• (1990).
- [10] McConnel, H. M., Structures and transitions in lipid monolayers at the air water interface, Annu. Rev. Phys. Chem., 42, 171-205 (1991).
- [11] Knobler, C. M. and Desai, R. C., Phase transition in monolayers, Annu. Rev. Phys. Chem., 43, 207 236 (1991).
- [12] Riviere, S., Henon, S., Meunier, J., Schwartz, D. K., Tsao, M. W. and Knobler, C. M., Texture and phase transitions in Langmuir monolayers of fatty acids a comparative Brewster angle microscope and polarized fluorescence study, J. Chem. Phys., 101, 10045 10051 (1994).
- [13] Knobler, C. and Schwartz D., Langmuir and self-assembled monolayers, Curr. Opinion Colloid Interface Sci., 4, 46-51 (1999).