References

[1] Afzal O. S., andVaswani N., and Tannenbaum A., “Object Tracking: A Survey”. IEEE, 2009, No. 6, pp. 4244–4249.

[2] AvidanS., Support vector tracking. Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition,2001, Vol. 1, pp. 184–191.

[3] Bibby C. and Reid I., Robust real-time visual tracking using pixel-wise posteriors.In ECCV ’08: Proceedings of the 10th European Conference on Computer Vision 2008,pp. 831–844.

[4]BlakeM. and Jepson A.,Eigentracking: Robust matching and tracking of articulatedobjects using a view-based representation. International Journal of Computer Vision,1998, 26(1): pp. 63–84.

[5]BradskiG.R.,"Computer vision face tracking for use in a perceptual user interface", Intel tech. J., May 1998, Vol. 2, Issue 2, pp. 12-21.

[6] Changjiang Yang, RamaniDuraiswami, Larry Davis: Efficient mean-shift tracking via a new similarity measure in Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR ’05).

[7] Collins R. T., Mean-shift blob tracking through scale space. Proceedings of IEEEConference on Computer Vision and Pattern Recognition,June 2003, Vol.2, pp. 234–240.

[8] Comaniciu D., Ramesh V., and Meer P., “Kernel-based object tracking,” IEEE Transactions on Pattern Analysis and Machine Intelligence,2003, pp. 564–575.

[9] Computer Vision Face Tracking For Use in a Perceptual User Interface, Intel Technology Journal, 1998, No.Q 2.

[10]Fukunaga K. and Hostetler L., The estimation of the gradient of a density function,with applications in pattern recognition. IEEE Transactions on Information Theory,January 1975, 21(1): pp. 32–40.

[11]Greg Welch, Gary Bishop, SIGGRAPH,An Introduction to the Kalman Filter 2001.

[12] Hu W., Tan T., Wang L., andMaybank S., “A survey on visual surveillance of object motion and behaviors,” IEEE Transactions on Systems, Man, and Cybernetics, Part C:Applications and Reviews, 2004Vol. 34, no. 3, pp. 334–352.

[13] JamilDrareni, Sebastien Roy,A Simple Oriented Mean-Shift Algorithm for Tracking.ICIAR 2007: pp. 558-568.

[14] Jepson A.,Fleet D., andEl-Maraghi T., “Robust online appearance models for visual tracking,” IEEE Transactions on Pattern Analysis and Machine Intelligence, 2003, pp.1296–1311.

[15] Kadir T. and Brady M., Non-parametric estimation of probability distributions fromsampled signals. Technical Report 2283/05, University of Oxford, 2005.

[16]Kalman R. E., “A new approach to linear filtering and prediction problems,”,Transactions of the ASME–Journal of Basic Engineering 82, Series D,1960, pp. 35-45.

[17] LindebergT.,Scale-space theory in computer vision, Monograph 1994.

[18] Lucas B., Kanade T. et al., “An iterative image registration Technique with an application to stereo vision,” International joint conference on artificial intelligence, Vol. 3. Citeseer, 1981, pp. 674–679.

[19] Pu B.,Liang S., Xie Y., Yi Z., and Heng P., “Video facial feature tracking with enhanced asm and predicted meanshift,” Second International Conference on Computer Modeling and Simulation. IEEE , 2010, pp. 151–155.

[20]Reid I. D. and Murray D.W., Tracking corner clusters using affine structure. IEEE International Conference on Computer Vision,May 1993 pp. 76–83.

[21] Sato K. andAggarwal J. K., Temporal spatio-velocity transform and its application totracking and interaction. Computer Vision and Image Understanding,2004 96(2): pp. 100–128.

[22] Shi J. andTomasi C.,"Good Features to Track,"9th IEEE Conference on Computer Vision and Pattern Recognition.(June 1994).

[23]TaoH., Sawhney H., andKumar R., Object tracking with bayesian estimation of dynamic layer representations. IEEE Transactions on Pattern Analysis and MachineIntelligence,2002, 24(1): pp. 75–89.

[24]Veenman C. J., ReindersM. J. T., IEEE Transactions on Pattern Analysis and Machine Intelligence, 2001, 23(1): pp. 54–72.

[25]WilliamsO., Blake A., and CipollaR., Sparse bayesian learning for efficient visual tracking. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2005,27(8):pp. 1292–1304.

[26]Yilmaz A.,Li X., andShah M., Contour-based object tracking with occlusion handling in video acquired using mobile cameras. Pattern Analysis and Machine Intelligence, IEEE Transactions,2004,26(11):pp1531–1536.

[27] YilmazA., JavedO., and Shah M., Object tracking: A survey. ACM Computing Surveys,2006, 38(4): p 13

APPENDIX

Program 1: Kalman Initialization

```
// Kalman initialization
CvKalman* InitializeKalman(CvKalman* kalman)
{
      const float A[] = {1,0,1,0,
      0,1,0,1,
      0,0,1,0,
      0,0,0,1};
      kalman = cvCreateKalman( 4, 2, 0 );
      memcpy( kalman->transition_matrix->data.fl, A, sizeof(A));//A
      cvSetIdentity( kalman->measurement_matrix, cvScalarAll(1) );//H
      cvSetIdentity( kalman->process_noise_cov, cvScalarAll(1e-5)
);//Q w ;
      cvSetIdentity( kalman->measurement_noise_cov, cvScalarAll(1e-1)
);//R v
      cvSetIdentity( kalman->error_cov_post, cvScalarAll(1));//P
      return kalman;
}
```

Program 2: Get Current State and Measurement State

```
void GetCurrentState( CvKalman* kalman, CvPoint point1, CvPoint
point2)
{
      float input[4] = {point2.x, point2.y, point2.x-point1.x,
point2.y-point1.y};//currentstate
      memcpy( kalman->state_post->data.fl, input, sizeof(input));
}

CvMat* GetMeasurement(CvMat* mat,CvPoint point1, CvPoint point2)
{
      float input[4] = {point2.x, point2.y, point2.x-point1.x,
point2.y-point1.y};
      memcpy( mat->data.fl, input, sizeof(input));
      return mat;
}
```

Program 3: Select Object

```
if( select_object )
{
      selection.x = MIN(x,origin.x);
      selection.y = MIN(y,origin.y);
      selection.width = selection.x + CV_IABS(x - origin.x);
      selection.height = selection.y + CV_IABS(y - origin.y);

      selection.x = MAX( selection.x, 0 );
      selection.y = MAX( selection.y, 0 );
      selection.width = MIN( selection.width, image->width );
      selection.height = MIN( selection.height, image->height );
      selection.width -= selection.x;
      selection.height -= selection.y;
}
```

```
switch( event )
{
      case CV_EVENT_LBUTTONDOWN:
      origin = cvPoint(x,y);
      selection = cvRect(x,y,0,0);
      select_object = 1;
      break;
      case CV_EVENT_LBUTTONUP:
      select_object = 0;
      if( selection.width > 0 && selection.height > 0 )
      track_object = -1;
      origin_box=selection;
}
```

Program 4: Camshift and Kalman Filter

```
//Predict the target frame position (x, y, vx, vy),
            prediction = cvKalmanPredict( kalman, 0 );
            predictpoint = calc_point(prediction->data.fl);
            //Predict the rectangular box
            track_window = cvRect(predictpoint.x -
track_window.width/2, predictpoint.y - track_window.height/2,
track_window.width, track_window.height);
            track_window=my_ChangeRect(cvRect(0,0,frame->width,frame-
>height),track_window);
            //Only calculate the projection of the target around
            search_window = cvRect(track_window.x - region,
track_window.y - region, track_window.width + 2*region,
track_window.height + 2*region);
            //Correction search_window, to within the range of image
            search_window=my_ChangeRect(cvRect(0,0,frame-
>width,frame->height),search_window);
            cvSetImageROI( hue, search_window );
            cvSetImageROI( mask, search_window );
            cvSetImageROI( backproject, search_window );
            cvCalcBackProject( &hue, backproject, hist );
            cvAnd( backproject, mask, backproject, 0 );
            //Because the set _1ROI, so updates track_window
            track_window = cvRect(region, region, track_window.width,
track_window.height);
            //Calling CAMSHIFT algorithm module
            cvCamShift( backproject, track_window, cvTermCriteria(
CV_TERMCRIT_EPS | CV_TERMCRIT_ITER, 2, 1 ), &track_comp);
            //cvMeanShift( backproject, track_window, cvTermCriteria(
CV_TERMCRIT_EPS | CV_TERMCRIT_ITER, 10, 1 ),&track_comp);


// The actual coordinates of the centroid
      measurepoint = cvPoint(track_window.x + track_window.width/2,
track_window.y + track_window.height/2 );

      realposition->data.fl[0]=measurepoint.x;
      realposition->data.fl[1]=measurepoint.y;
      realposition->data.fl[2]=measurepoint.x - lastpoint.x;
      realposition->data.fl[3]=measurepoint.y - lastpoint.y;
      lastpoint = measurepoint;//keep the current real position

//Calculate the actual measurement is the current x, y
```

```
cvMatMulAdd( kalman->measurement_matrix/*2x4*/,
realposition/*4x1*/,/*measurementstate*/ 0, measurement );
            cvKalmanCorrect( kalman, measurement );

cvKalmanCorrect( kalman, measurement );
```