Research on the Algorithm of Concave Fillet Auto Decreasing in Panel Die

Shan-Hui Zhang[1], Wei Xu[2], Wei Wei[2]

1 *School of Control Science and Engineering*
Shandong University
Jinan, 250061, China
2 Shandong Shanda Hoteam Soft Co., Ltd
Jinan, 250101, China

***Abstract* — Such problems as poor quality and low processing efficiency of fillet surface deformation can easily appear during the process of adopting CAD software to decrease the fillet in panel die. In order to solve above problems, a fillet automatic decreasing algorithm based on surface-surface fillet creation is proposed in this article. The fillet surfaces can be automatically decreased by batch through a series of algorithms including automatic fillet search algorithm, surface-edge search algorithm, surface lengthening algorithm, surface-surface fillet smoothening algorithm, variable radius fillet algorithm, etc., and meanwhile relevant research and development are carried out in domestic CAD software, thus to significantly improve fillet design efficiency. Furthermore, the application of this algorithm in a vehicle door surface fillet decreasing case shows that this algorithm can improve fillet decreasing efficiency and the quality of the deformed fillet surface, thus to meet the efficient and flexible die surface design requirements of die enterprises.**

Keywords - panel die; fillet decreasing; surface-surface fillet creation; variable radius

I. INTRODUCTION

The fillet of large panel die can significantly influence the final quality and the die assembly rate of a stamping part, so the method of decreasing concave fillet radius is usually adopted by die enterprises and the fillet decreasing process also becomes an indispensible design link. The traditional method of arranging bench workers to manually grind concave fillets has the features of large labor intensity, low efficiency and poor quality guarantee. At present, CAD software is universally adopted to decrease the concave fillet radius [1] in the molded surface before numerical control machining, and the available two methods are as follows: (1) select all concave fillets and drop down by a certain distance along the negative Z-axis direction [2]; (2) adopt surface deformation function to decrease concave fillets and keep convex fillets unchanged in order to make up-down joint have certain gap and meanwhile make product molded towards convex die. In present design method, the fillets only can be processed one by one, so such problems as tedious operations, long design cycle and side fillet processing failure will be caused, the subsequent grinding workload will be also increased, and the fillet surface generated thereby will have poor quality [3].

At present, a small amount of software can provide fillet solutions, and only two types of foreign software, namely Tebis and NX, can provide one-by-one fillet processing method, but such software has low design efficiency and fairly high price [4].

Therefore, in order to improve the design and manufacturing efficiency of large panel die and the fillet quality, it is greatly necessary to develop and research efficient and flexible fillet processing function on the basis of domestic three-dimensional CAD systems [5-6].

II. DESIGN AND ALGORITHM OF FILLET AUTOMATIC DECREASING

The fillet decreasing function is mainly used to decrease the fillet surface relatively to the shape of the original fillet surface according [7-9] to the parameterized and non-parameterized fillet surface data. During fillet processing, it is necessary to analyze and identify fillet surfaces and then decrease these fillet surfaces, wherein the latter is the core of the whole processing procedure [10].

In CAD system, several fillet decreasing methods are available, and the most common methods includes edge fillet creation and surface-surface fillet creation, wherein the edge fillet creation method can easily generate wave-like fillet surface with poor quality [11]; for the surface-surface fillet creation method, the processing logics for the lengthened surface and the clipped surface are relatively complicated [12]. Through careful analysis, evaluation and verification, the surface-surface fillet creation method is finally selected and adopted, and meanwhile a fillet automatic decreasing algorithm based on surface-surface fillet creation is designed in this article for this surface-surface fillet creation, and this algorithm aims at analyzing the processing logics for the lengthened surface and the clipped surface in order to ensure the correct deformation of the fillet surface. The algorithm implementation process mainly includes the following five steps:

(1) Search fillet surface: according to the molded surface selected thereby, search and identify each surface and judge whether this surface is a fillet surface or not;

(2) Search surface edge: find the long side boundary line and R side boundary line of the fillet surface;

(3) Lengthen surface: lengthen the long boundary lines at the two sides of the fillet surface along the opposite direction of the tangent lines of two R side boundary lines, and find the two tangent planes;

(4) Smoothen surface-surface fillet: lengthen two tangent planes and smoothen surface-surface fillets, create new fillet surface and clip redundant surface;

(5) Process variable radius fillet: process the special variable radius fillets.

A. *Fillet Surface Searching*

The steps are as follows: designate a series of molded surfaces and then search satisfactory fillet surfaces among the molded surfaces. During the process of searching fillet surfaces, it is necessary to judge whether the surface is a fillet surface according to the geometrical information thereof. Obviously, this algorithm has wide application and can be applicable to parametric and non-parametric molded surfaces.

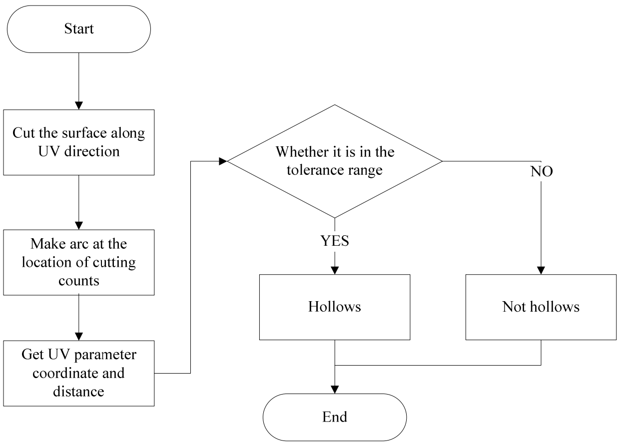


Figure 1. Fillet surface judgment process

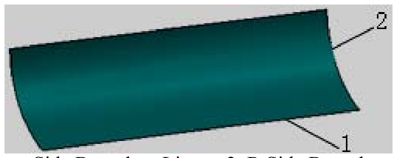
The process of judging whether the present surface is a fillet surface is as shown in Fig. 1. During fillet surface judgment process, it is necessary to cut the fillet surface along UV direction according to the input cutting counts, respectively take the coordinate values of the positions with the parameters of 0.25 and 0.75, and further calculate the distance to the nearest arc. If the distances along UV direction are all within the allowable tolerance range, then the present surface can be judged as a fillet surface; or else, the present surface is not a fillet surface.

In the fillet searching function, the key algorithm code of all single surfaces on the searched curved surface is as follows:

```
stat = TiInqComFac(&idCom1,&numfac1,&faceidP1);
    if (stat != 0){
        goto rtn;}
stat = TiInqComFac(&idCom2,&numfac2,&faceidP2);
    if (stat != 0){
        goto rtn;}
```

B. *Surface Edge Searching*

The process of searching surface edge aims at searching long side boundary line and R side boundary line of the fillet surface, thus to prepare for the processing of the lengthened surface. As shown in Fig. 2, after surface edge searching, it is necessary to lengthen the surface respectively from the long side boundary lines of the fillet surface along the opposite directions of the tangent lines of R side boundary lines to obtain two tangent planes, namely the lengthened surfaces.



1. Long Side Boundary Line 2. R Side Boundary Line

Figure 2. Long side and R side boundary line

The process of searching the long side boundary lines and R side boundary lines of a fillet surface is as shown in Fig. 3. The above boundary lines are determined by judging whether the curvature radius of the boundary lines of the fillet surface is consistent with the curvature radius of the fillet. If yes, the present boundary line is judged as an R side boundary line; if not, the present boundary line is a long side boundary line.

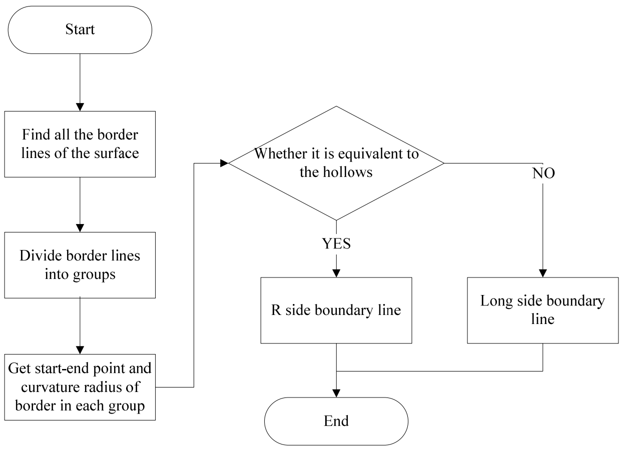


Figure 3. Surface edge searching process

Part of the key codes for searching surface edge is as shown in Table 1.

TABLE I PART OF CODES FOR SEARCHING SURFACE EDGE

```
/*Obtain boundary line*/
        stat = TiInqComFac(&idCom1,&numfac1,&faceidP1);
        if (stat != 0){
                goto rtn;
/*Obtain long side boundary line*/
        stat = yxss00( outsrfcanonAP1, outsrfcanonAP2, 0, 1, tolAP, &headSPI );
        if (stat != 0){
                goto rtn;
        }
/* Obtain R side boundary line*/
        cnum = VMYHCMNNUM(VMYHGPSCMN(headSPI)+1);
        }
```

TABLE II PART OF SURFACE LENGTHENING CODE

```
/* Lengthen fillet surface*/
        stat = VqlLftExSf ( srfcanonAP1, uvflgA, exlngA, &outsrfcanonAP1, tolAP );
        if (stat != 0){
                goto rtn;
        }
/* Find the intersection line*/
        stat = yxss00( outsrfcanonAP1, outsrfcanonAP2, 0, 1, tolAP, &headSPI );
        if (stat != 0){
                goto rtn;
        }
```

TABLE III SURFACE-SURFACE FILLET SMOOTHENING INTERFACE

VPLFLTM00(	
VtLfltIdent *id0SPI,	// The 1st surface information
VtLfltIdent *id1SPI,	// The 2nd surface information
INT *guidFP,	// Whether there is a guide line
VtLfltIdent *gcrvidSPI,	// Guide line information
VtLfltArcMd *arcmtdSPI,	//Arc creation method
VtLfltRadIf *radinfSPI,	// Radius information
VtLfltCreRg *crerngSPI,	// Fillet surface creation range
VtLfltMOptIf *optinfSPI,	// Option information
DOUBLE *radP,	//Certain radius value
INT *mnumP)	// Variable radius R value

TABLE IV VARIABLE RADIUS FILLET PROCESSING INTERFACE

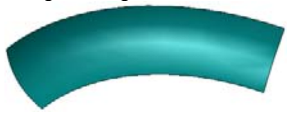
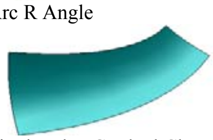
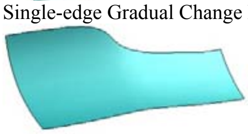
INT VqlGetFltPntRad(	
YSf *tansurCD,	//Tangent plane constraint
YCv *crvCP,	//Public boundary line of tangent plane
INT divNum,	//Number of partitions of the public boundary line of the tangent plane
YSf *fltsurCD,	// Fillet data
INT uvflag)	// Fillet direction sign

TABLE V PART OF VARIABLE RADIUS FILLET PROCESSING CODES

```
/*Calculate the tangent vector of the point on the boundary line*/
        if( (ystat = yvcv ( cvCP1->bcrrv_p, pt, 1, cvalA ) ) != (INT)0 ){
                stat = VEYRTN;
                goto rtn;
        }
/* Calculate the tangent vector of the point on the plane along UV direction*/
        if( (ystat = yvsf ( sfCP1->bsrf_p, prm1A, 1, svalA, nrmA ) ) != (INT)0 ){
                stat = VEYRTN;
                goto rtn;
                }
```

```
    /* Compare the angle between the tangent vector of the boundary line and UV and the angle between the tangent
vectors along two directions*/
            if( (zstat = zqvt ( D33D, tngA, uvecA, &inner_u ) ) != (INT)0 ){
                        stat = VEYRTN;
                        goto rtn;
            }
            if( (zstat = zqvt ( D33D, tngA, &inner_v ) ) != (INT)0 ){
                        stat = VEYRTN;
                        goto rtn;
                        }
        }
```

TABLE VI ALL FILLET TYPES

Automatic Decreasing R Angle	Manually Adjusted R Angle
 Straight R Angle  Arc R Angle  Single-edge Gradual Change R Angle  Double-edge Gradual Change R Angle	 Confluence R Angle

C. Surface Lengthening

Lengthen the surface along the long side boundary lines and the opposite directions of the tangent lines of R side boundary lines to create two planes, namely the lengthened surfaces. The main implementation algorithm is as follows:

(1) Search the neighboring R side boundary line according to the ling side boundary line and then find the direction of the tangent line thereof;

(2) Find the lengthened boundary lines at the two sides according to the direction of the tangent line (opposite direction) and the lengthening distance;

(3) Calculate the average direction of the lengthening directions of the two boundary lines and find the boundary line at the middle of the two side boundary lines according to the average direction and the lengthening distance;

(4) Repeat step (1) and step (3) to create the boundary line of each part;

(5) Finally, create the surface according to the track of all boundary lines.

Adopt primitive function of the surface lengthening function and lengthen the surface along the direction of the tangent line of the neighboring edge. Part of the key codes is as shown in Table 2.

D. Surface-Surface Fillet Smoothening

Surface-surface fillet smoothening means to create new fillet surface according to the lengthened surfaces, the preset fillet radius and the surface-surface fillet smoothening method. This function is achieved mainly by calling the surface-surface fillet smoothening command interface (VPLFLTM00). The primitive function of surface-surface fillet smoothening is described in the interface as shown in Table 3.

E. Variable Radius Fillet Processing

Reversed gradual change may appear during the process of adopting fillet decreasing function to adjust variable radius fillet. For example, before variable radius fillet adjustment, the radius is R[3] ={10.0，8.0，6.0}; after the radius is decreased by 20%, radius shall be R[3] ={8.0，6.4，4.8}[9]; but the initial algorithm is not rigorous for calculating the interpolation radius of the interpolation point in the processing procedure, thus causing the following two problems:

(1) After adjustment, the gradual change direction of the fillet is reversed, namely R’[3] ={4.8， 6.4， 8.0}, thus

causing fillet adjustment failure or divergence with other fillets after adjustment.

(2) The relatively large basal surface of the fillet will cause the interpolation point and the interpolation radius calculated thereby not to be accurate enough, namely R'[3]={ 8.3， 6.8， 5.0}.

Due to the above problems, the above interpolation algorithm is optimized and improved in order to simplify interpolation point processing procedure. Specifically, the interpolation points are directly obtained on the boundary lines, thus to improve the accuracy of the interpolation radius calculation of the variable radius fillet and accelerate the processing efficiency as well as improve the generation quality. The implementation algorithm is as follows:

(1) Create and clip the tangent planes, and then obtain the public boundary line;

(2) Set sampling points on the boundary line, namely the interpolation points for creating variable radius fillets;

(3) Take each sampling point as the starting point of the isoparametric line, then calculate the end point of the corresponding isoparametric line on the tangent plane, wherein the end point is corresponding to a certain point on the boundary line of the fillet surface;

(4) Obtain the end point of the tangent surface through calculation and then calculate the radius of the isoparametric line of the fillet surface, wherein this radius is namely the interpolation radius corresponding to the interpolation point.

The interface design for calculating the interpolation point of the variable radius fillet and the interpolation radius is as shown in Table 4.

For the variable radius fillet decreasing scenario, in order to improve fillet quality, the interpolation point is adopted at the end point of the corresponding isoparametric line on the tangent plane for calculation. Part of the codes is as shown in Table 5

III. Fillet Automatic Decreasing Function And Implementation

A. Variational Fractal Dimension Calculation Model

Among several fractal dimension calculation methods, variational fractal dimension calculation method has simple solution and small error. In this article, the variational fractal dimension calculation model is adopted to calculate the fractal dimension point by point so as to convert the image gray space into fractal dimension space.

Under the assumption that set *S* is composed of sampling points *z(xi，yi)*, namely *z(xi，yi)* ∈*S*, and is obtained from continuous function sampling, and the condition $x \in [a, b]$ and $y \in [c, d]$ can be met, then for a real number *t*, the stage difference value of point t in the neighborhood of (*x*，*y*) can be defined as follows:

$$ocs(x,y,t)=\max\left[z\left(x^{'},y^{'}\right)-z\left(x^{''},y^{''}\right)\right] \quad (1)$$

In the above formula, $|x^{'}-x|\le t$, $|x^{''}-x|\le t$, $|y^{'}-y|\le t$, $|y^{''}-y|\le t$; if $\Delta(t)=\frac{1}{(b-a)(d-c)}\int_a^b\int_c^d osc(x,y,t)dxdy$, then the fractal dimension of the fractal surface is as follows:

$$Ds=\lim_{t\to 0}\left(3-\lg\Delta(t)/\lg t\right) \quad (2)$$

B. Direction Fractal Dimension Calculation Model

In order to effectively measure the mutation features and the direction features of the boundary points and well detect the peripheral points, the author has defined the direction information measurement and substantially adopted the grey mutation along different directions to detect and correctly classify the peripheral points. In this article, we refer to the feature description for the peripheral points in this definition and accordingly define the direction fractal dimension measurement. However, due to the particularity of the optimal direction angle image created in this article, we need to consider the following two problems before adopting this definition to detect the boundary points: (1) angle has particularity: we cannot directly judge the angle mutation degree according to the absolute value of the direction angle; in other words, the first problem is period equivalence of two angles and distance judgment; (2) for the "local optimal cutting direction" of each point after calculation, according to the previous definition, if the tangent direction of the point on the cutting track is identical or opposite to a certain local optimal cutting direction at this point, then the cutting width corresponding to this point is maximum, namely optimal cutting track; in other words, the second problem is the optimal direction equivalence of angle. Based on the above reasons, it is necessary to give the new description method of data point set and consider the above two problems to define the direction fractal dimension measurement.

Definition 1: if θ is an optimal direction angle of a certain point, then $\Theta=\{\theta+k\pi, k\in Z\}$ is called as the optimal period equivalence angle set of θ.

Definition 2: for two points $\theta_{i,j}$ and $\theta_{i',j'}$ in the optimal direction angle picture, the optimal direction equivalence angle set is defined as follows:

$$\left|\theta_{i,j}-\theta_{i^{'},j^{'}}\right|=\begin{cases}\left|\theta_{i,j}-\theta_{i^{'},j^{'}}\right| & \left|\theta_{i,j}-\theta_{i^{'},j^{'}}\right|\le\frac{\pi}{2}\\ \pi-\left|\theta_{i,j}-\theta_{i^{'},j^{'}}\right| & \left|\theta_{i,j}-\theta_{i^{'},j^{'}}\right|>\frac{\pi}{2}\end{cases} \quad (3)$$

This definition is given for the purpose of correctly judging the distance between such angles as 175^o and 3^o in the direction angle neighborhood picture so as to realize accurate measurement [34].

Through the definition for the two equivalence angle sets, we can preprocess the optimal direction angle neighborhood picture so as to convert all the direction angles to the interval of $[0,\pi)$.

Definition 3: the direction fractal dimension measurement is defined as follows:

If the coordinate of the present pixel point is (i, j), its neighborhood is $N(i, j)$ and l_θ is a straight line passing through the central point at the angle of θ_i (including eight directions, namely $0, \pi/4, \pi/2, 3\pi/4, \pi, 5\pi/4, 3\pi/2, 7\pi/4$), then the direction fractal dimension measurement $\Delta D_{i,j}$ of point (i, j) is defined as:

$$\Delta D_{i,j} = \max_{0^\circ \le \theta \le 360^\circ} \Delta d_{\theta i} \quad (4)$$

$$\Delta d_{\theta_i} = \left| d_{\theta i} - d'_{\theta i} \right| \quad (5)$$

In the formula, d_{θ_i} and d'_{θ_i} respectively denote the variational fractal dimension corresponding to the pixel points of the optimal direction angles at points (i, j) and (i', j') along the direction with θ_i as the axis. If the present neighborhood has the edge passing through the central point, then the peripheral point has directivity and $\Delta D_{i,j}$ can have maximum value, namely: the direction fractal dimension measurement of the peripheral point is relatively large.

In this way, the maximum fractal dimension difference of the central point of the neighborhood is substantially provided through above definitions for effectively extracting the boundary points of the picture.

C. *Algorithm Implementation*

Concave R angle preprocessing function has been realized in domestic three-dimensional CAD system and has also been applied in six large automobile panel die design and manufacturing enterprises. In the example of adjusting the fillet on the deformed vehicle door die surface: firstly, search all concave fillet surfaces among the vehicle door die surfaces according to the preset parameters, as shown in Fig. 5; then, set relevant parameter for decreasing the fillet radius proportionally by 20%, namely automatically decreasing all fillet surfaces by batch, as shown in Fig. 6. According to the field debugging and verification, we can know that users think highly of concave R angle preprocessing function and greatly accept the practicability and the accuracy of the same. According to the analysis of the whole design process, users' workload for amending fillet can be reduced by about 80%.

Through concave R angle preprocessing function, the change effect of the decreased fillet can be clearly viewed. The comparison between the fillet decreased to R10 and the fillet decreased to R8 is as shown in Fig. 6.

According to the internal test and the field test carried out by users for concave R angle preprocessing function, we can know that concave R angle preprocessing function is applicable to all fillets, as shown in Table 6. Therein, the automatic decreasing R angles include straight R angle, fillet R angle, single-edge gradual change R angel and double-edge gradual change R angle; manually adjusted R angle includes confluence R angle.

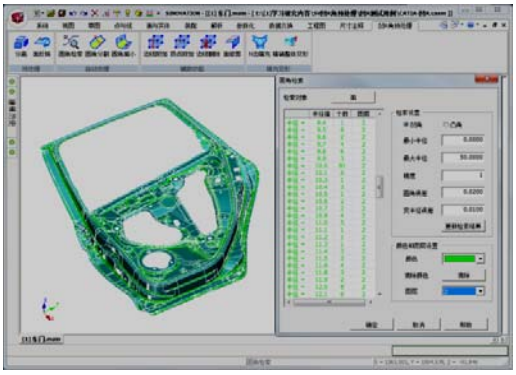


Figure 4. Fillet Searching

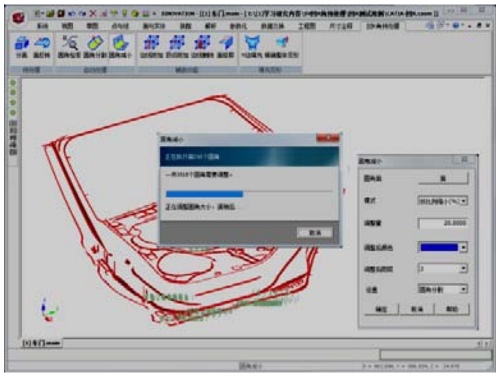
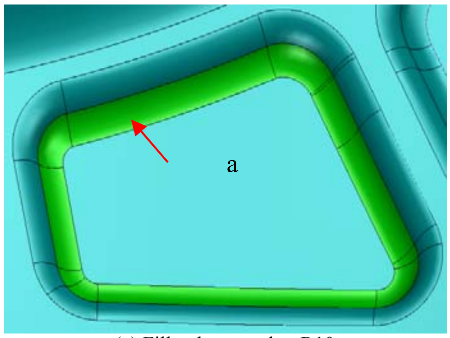
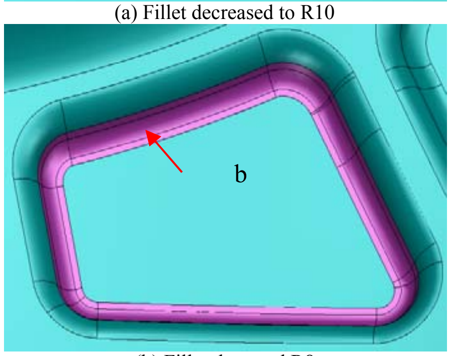


Figure 5. Fillet decreasing



(a) Fillet decreased to R10



(b) Fillet dcreased R8

Figure 6. Fillet decreasing contrast effect.

DOI 10.2013/IJSSST.a.17.05.11 ISSN: 1473-804x online, 1473-8031 print

IV. CONCLUSION

In order to solve such problems as poor quality and low processing efficiency of fillet surface deformation during the process of adopting CAD software to decrease the fillet in panel die, a fillet automatic decreasing algorithm based on surface-surface filet creation is proposed in this article to automatically decrease fillet surfaces by batch. This algorithm has been developed and researched in domestic CAD software for relevant function realization. After the application of the algorithm in the fillet decreasing in panel die, the fillet design efficiency has been significantly improved. Compared with traditional method, the workload for amending fillet is reduced by at least 80% and the decreased fillet surface has improved quality, thus to meet the requirement for the molded surface design of die. Therefore, users think highly of this algorithm.

ACKNOWLEDGMENT

This work is supported by the Special Program for International S&T Cooperation Projects of China under Grant No. 2011DFB11490.

REFERENCES

[1] Alex Tek, Benoist Laurent, Marc Piuzzi, Zhihan Lu, Matthieu Chavent, Marc Baaden, Olivier Delalande et al. "Advances in Human-Protein Interaction-Interactive and Immersive Molecular Simulations". InTech, 2012.

[2] Degui Zeng, Yishuang Geng, "Content distribution mechanism in mobile P2P network", Journal of Networks, Vol. 9, No. 5, pp. 1229-1236, 2014.

[3] Dingde Jiang, Zhengzheng Xu, Peng Zhang, et al., "A transform domain-based anomaly detection approach to network-wide traffic", Journal of Network and Computer Applications, vol. 40, pp.292-306, 2014.

[4] Dingde Jiang, Zhengzheng Xu, Wenpan Li, Zhenhua Chen, "Network Coding-Based Energy-Efficient Multicast Routing Algorithm for Multi-hop Wireless Networks", Journal of Systems and Software, Vo. 12, 2015.

[5] Gan Yan, Yuxiang Lv, Qiyin Wang, Yishuang Geng, "Routing algorithm based on delay rate in wireless cognitive radio network", Journal of Networks, Vol. 9, No. 4, pp. 948-955, 2014.

[6] Guanqun Bao, Liang Mi, Yishuang Geng, Kaveh Pahlavan, "A computer vision based speed estimation technique for localiz ing the wireless capsule endoscope inside small intestine", 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), Aug. 2014

[7] Guanqun Bao, Liang Mi, Yishuang Geng, Mingda Zhou, Kaveh Pahlavan, "A video-based speed estimation technique for localizing the wireless capsule endoscope inside gastrointestinal tract", 2014 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), Aug. 2014

[8] Guanxiong Liu, Yishuang Geng, Kaveh Pahlavan, "Effects of calibration RFID tags on performance of inertial navigation in indoor environment", 2015 International Conference on Computing, Networking and Communications (ICNC), Feb. 2015

[9] Jie He, Yishuang Geng, Fei Liu, Cheng Xu, "CC-KF: Enhanced TOA Performance in Multipath and NLOS Indoor Extreme Environment", IEEE Sensor Journal, Vol. 14, No. 11, pp. 3766-3774, 2014

[10] Jie He, Yishuang Geng, Yadong Wan, Shen Li, Kaveh Pahlavan, "A cyber physical test-bed for virtualization of RF access environment for body sensor network", IEEE Sensor Journal, Vol. 13, No. 10, pp. 3826-3836, 2013.

[11] Liguo Zhang, Binghang He, Jianguo Sun, Mingzhu Lai, Zhihan Lv. "Double Image Multi-Encryption Algorithm based on Fractional Chaotic Time Series". Journal of Computational and Theoretical Nanoscience. 2016.

[12] Mingda Zhou, Guanqun Bao, Yishuang Geng, Bader Alkandari, Xiaoxi Li, "Polyp detection and radius measurement in small intestine using video capsule endoscopy", 2014 7th International Conference on Biomedical Engineering and Informatics (BMEI), Oct. 2014